ESD-TR-77-217

Technical Note

1977-22

D. P. White

A Time Diversity Coding Experiment for a UHF/VHF Satellite Channel with Scintillation: Equipment Description

1 September 1977

Prepared for the Department of the Navy
under Electronic Systems Division Contract F19628-76-C-0002 by

Lincoln Laboratory

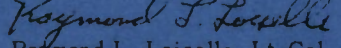
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LEXINGTON, MASSACHUSETTS

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. The work was sponsored by the Department of the Navy under Air Force Contract F19628-76-C-0002.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER



Raymond L. Loiselle, Lt. Col., USAF
Chief, ESD Lincoln Laboratory Project Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

A TIME DIVERSITY CODING EXPERIMENT FOR A UHF/VHF SATELLITE CHANNEL WITH SCINTILLATION: EQUIPMENT DESCRIPTION

D. P. WHITE
Group 66

TECHNICAL NOTE 1977-22

1 SEPTEMBER 1977

LEXINGTON MASSACHUSETTS

ABSTRACT

Satellite communications in the UHF and VHF bands has at various times and in certain receiver locations on the earth's surface been subject to moderate to severe degradation as a result of scintillation. The affected receiver areas are confined primarily to the equatorial regions and to a lesser extent over the sub-auroral to polar cap areas. For the equatorial region (more specifically, a band centered on the geomagnetic equator) the scintillation is largely a nighttime phenomenon beginning shortly after local sunrise and diminishing after midnight.

In response to the potential problem on the Navy's UHF fleet broadcast channel we have conducted a study on the communication alternatives as well as an extensive field experiment. Time diversity was identified as the only acceptable means of improving communications efficiency. Both threshold processing and coded modulation schemes were considered. Our analyses and computer simulations showed that a combination of convolutional encoding and interleaving of the transmitted data coupled with deinterleaving and Viterbi decoding at the receiver gives a major improvement in error protection performance. The field experiment was conducted between Hawaii and Guam over the Pacific Gapfiller (Marisat) satellite from August 1976 through January 1977.

We assembled a modest size system which was interfaced with the Navy's broadcast transmitter facility and another which was interfaced with the standard SSR-1 receiver. The experiment compared the error rate performance for systematic and nonsystematic convolutional codes and for coherent as well as differentially coherent detection. Three different code rates (1/2, 2/3, 3/4) were used. This first report contains the operational configuration and a future report the detailed results.

TABLE OF CONTENTS

I. OVERVIEW

For a VHF or UHF satellite channel in either the equatorial or auroral regions, the ionospheric disturbances that cause "spread F" can also cause significant signal amplitude fading (scintillation). Inasmuch as the Navy will be a heavy UHF user, it is advisable to consider communications techniques which mitigate against the more severe fading conditions. One link of particular importance is the fleet broadcast. Providing protection to this link under scintillating channel conditions is the subject of this report.

Two recent studies have suggested possible approaches to the problem. R. K. Crane [1,6] has analyzed a significant quantity of UHF and VHF scintillation data and has interpreted these data in a form that a communications specialist can utilize. In particular Crane demonstrates that neither polarization nor frequency diversity are useful techniques and that spatial diversity is not practical on a receiver platform of limited length such as a ship.

A. Coding Techniques

Time diversity appears to be the only practical means. Presented with the fading statistics, Bucher [2] investigated various time diversity techniques. Both threshold processing and interleaved/coded modulation schemes were analyzed. With threshold processing the received message is divided into segments many bits in length but shorter than a fade duration. The quality of each segment is compared to a threshold and the message segment is rejected if it does not meet the threshold. The ensuing message therefore has gaps which can be filled in a number of alternative ways. In the first method, multiple repeats of the message segments are always sent and the receiver pieces together the

segments and attempts to produce one which is error free. This technique is clearly inefficient in terms of capacity. In the second method only those message segments which were rejected are then requested for retransmission. This is more efficient in capacity but requires a two-way channel and complex control in multi-listener systems. A third method is to treat the gaps as erasures and provide an erasure filling algebraic coding scheme to reconstruct the message text. However, this scheme requires a fairly high degree of decoder complexity and is very sensitive to the channel statistics.

The interleaved/coded modulation schemes plus interleaving provide more flexibility in combatting a variety of link degradations and consequently our efforts were concentrated here.

It has been realized for some time that convolutional encoding and Viterbi decoding in conjunction with DPSK modulation is an efficient and reliable means of communicating over a satellite channel (in the absence of scintillation). Heller and Jacobs [3] have written a particularly lucid summary of the effects of certain choices of decoder parameters on an improvement in probability of bit error as a function of the energy per bit to noise density ratio (E_b/N_o, where N_o is the single sided noise spectral density). They concentrated on the case where the received signal consists of a constant signal plus additive noise. A number of simulations were performed where the additive component was assumed to be "white" Gaussian noise. The parameters investigated were the encoder constraint length K, path history length, degree of matched filter output quantization and imperfect carrier phase reference. Our intention was to perform a similar analysis with scintillation being included

as well. In the presence of scintillation Crane shows the fading is no worse than Rayleigh distributed and often is observed to be Rayleigh. Moreover an analysis of the sample fade duration probability distribution function showed most fade durations to be of the order of 1 second and less. With an appropriate interleaving scheme, the received data into a decoder can then be conservatively characterized for purposes of analysis as independent from channel symbol to channel symbol with Rayleigh amplitude statistics. Bucher computed the probability of error (P_E) curves versus $\overline{E}_b/N_o$ for a few representative algebraic block codes and compared them to some systematic convolutional codes with Viterbi decoding of modest constraint lengths (Fig. 1). The conclusion reached was that the convolutional codes generally offer better error rate characteristics. The Viterbi decoder for multiple code rates is relatively simple to build as compared to the equivalent algebraic decoders. Moreover it is easier to build into the Viterbi decoder provisions for utilizing the soft decision outputs of the matched filter. Bucher shows that the Viterbi decoder parameter choices are essentially the same for Rayleigh statistics as compared to those suggested in the results of Heller and Jacobs (which were based on a non-fading channel).

With convolutional codes there is a choice between systematic and non-systematic codes. The use of <u>systematic</u> convolutional codes is compatible with the present UHF fleet broadcast channel. The fleet broadcast channel consists of sixteen independent 75 bps data streams time division multiplexed to a single 1200 bps channel. One of the potential data streams is used to carry the frame synchronization. A systematic code will utilize one or more data streams to transmit the required segment of the broadcast message and use one

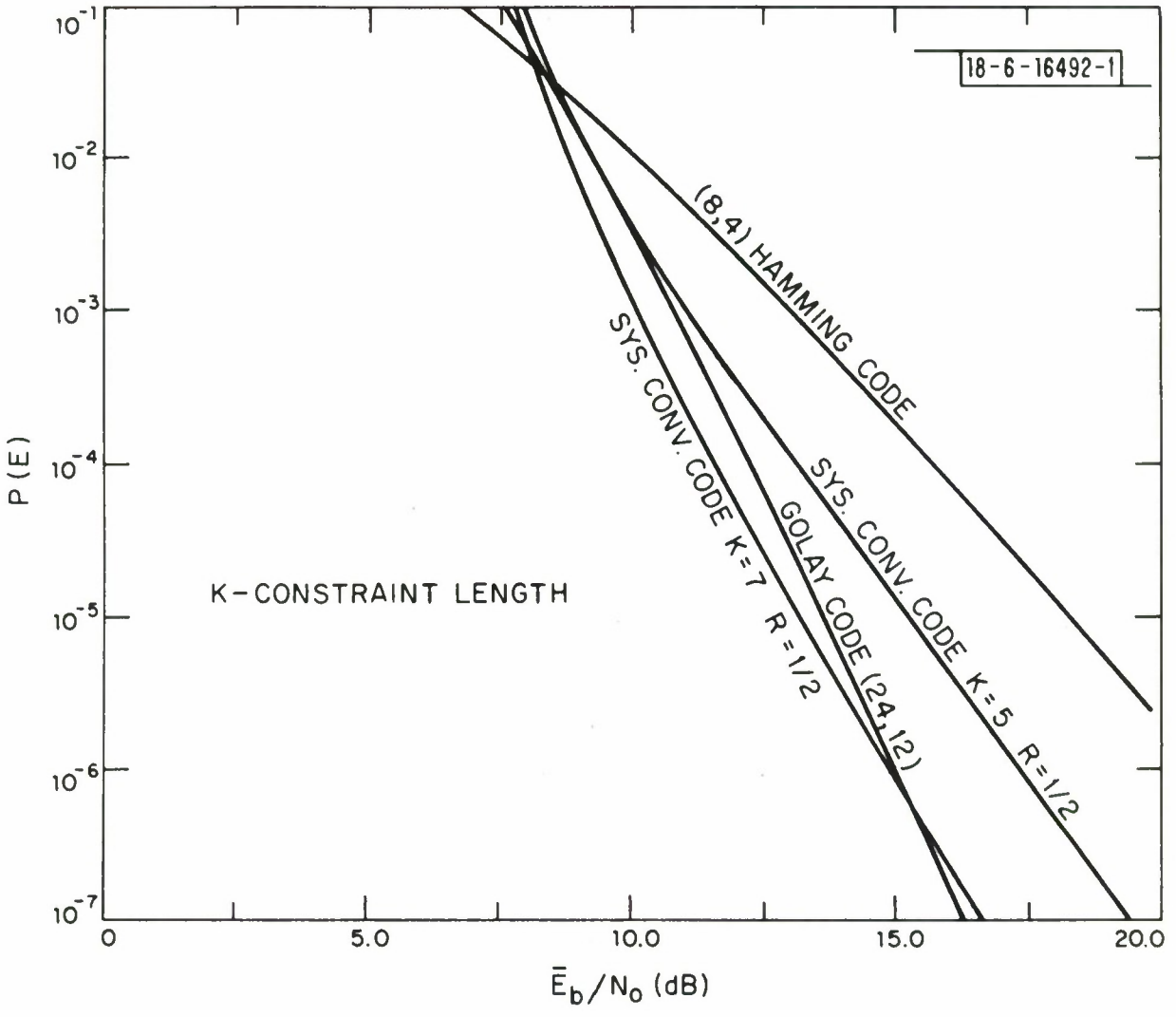


Fig. 1. Code performance comparison for Rayleigh fading (DPSK chip modulation, differential coherent detection and soft decisions. $\bar{E}_b$ is the average signal energy and N_0 the single sided noise density).

other data stream for the parity bits. For example a rate 1/2 systematic code will utilize one channel to transmit the actual 75 bps "message" and an additional 75 bps stream for the parity bits generated from the output of a convolutional encoder. Similarly, for a rate 2/3 systematic code, two 75 bps information streams are sent along with an additional 75 bps parity stream. It is evident that a fleet broadcast receiver without a special decoder can still receive the normal 75 bps message; however, it does lack error correcting capabilities. A more efficient system in terms of error correction capability would be a non-systematic code where all the transmitted data streams are encoded as well. Bucher shows a comparison of the performance gain for a rate 1/2 K=7 systematic and non-systematic convolutional code with Rayleigh feeding and with Viterbi decoding in Fig. 2. In the figure the term "pre-decoder" error rate refers to the error rate of a single information bit stream at the receiver bit prior to the decoder. The "post-decoder" error rate refers to the error rate of the same information bit sequence as measured after the Viterbi decoder. The main feature of this figure is that the probability of error curve for the pre-decoder data is relatively "flat" over a significant range of $\overline{E}_b/N_o$. As examples at the $\overline{E}_b/N_o$ ratios of 15 and 20 dB, the pre-decoder error rates are as large as 2×10^{-2} and 5×10^{-3}. By comparison it is seen that using either systematic or non-systematic coding will result in $P_E < 10^{-3}$ for values of $\overline{E}_b/N_o$ as small as 10 dB. As part of our experiment we are investigating both systematic and non-systematic codes. We recognize that the non-systematic codes provide better error protection although they are not compatible with the present fleet broadcast scheme unless all the fleet receivers on platforms required to monitor the protected information include a decoder.

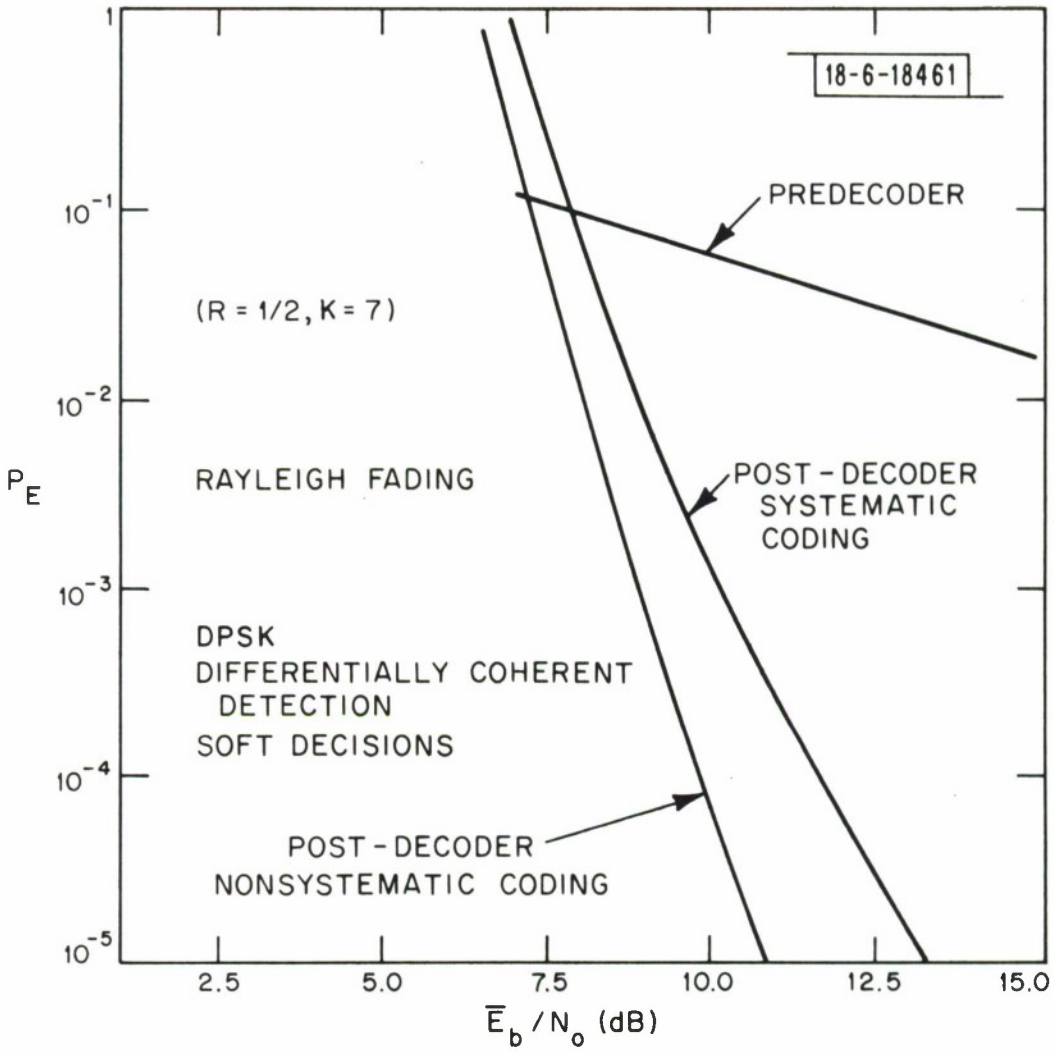


Fig. 2. Comparison of Error Probabilities for Systematic and Non-systematic Codes ($\overline{E}_b$ is the average received energy for information bit).

B. Experiment

Under the sponsorship of the Navy we used the fleet broadcast system as a test bed for our investigation. This meant in particular that we were allowed to use their spare transmitter multiplexer at a Communications Area Master Station (CAMS) and were provided with standard SSR-1 fleet broadcast receivers. Accordingly, our hardware effort was concerned with the development of four major subsystems. See Fig. 3 for identification of the various system components. First we fabricated an interleaving scheme for the systematic and non-systematic codes. Implicit in this is the design of an interleaver (as well as delay lines and deinterleaver) structure which has relatively simple synchronization requirements. Note that all the interleavers (and deinterleavers) associated both with the transmitter and the receiver must operate in synchronism and hence a periodic synchronization signal must be transmitted for this purpose. With fades of the order of 1 sec the interleaver memory requirements become large and standard RAM memories are logical choices. The multiplicity of interleavers require a design that exhibits a high degree of commonality in the RAM addressing scheme.

Second, we have looked into a simple means of implementing the Viterbi decoder at code rates 1/2, 2/3 and 3/4. Fortunately the data input rate is limited to 225 bps (for the highest code rate) and this allows use of a simple computer type structure, the advantage being a relatively small number of medium scale integration (MSI) chips (i.e., 80 chips). The details of this realization will be further developed in Section IV-C.

Third, we have made some analog circuit additions to the Navy's fleet broadcast SSR-1 receiver. The normal SSR-1 configuration has a Costas phase

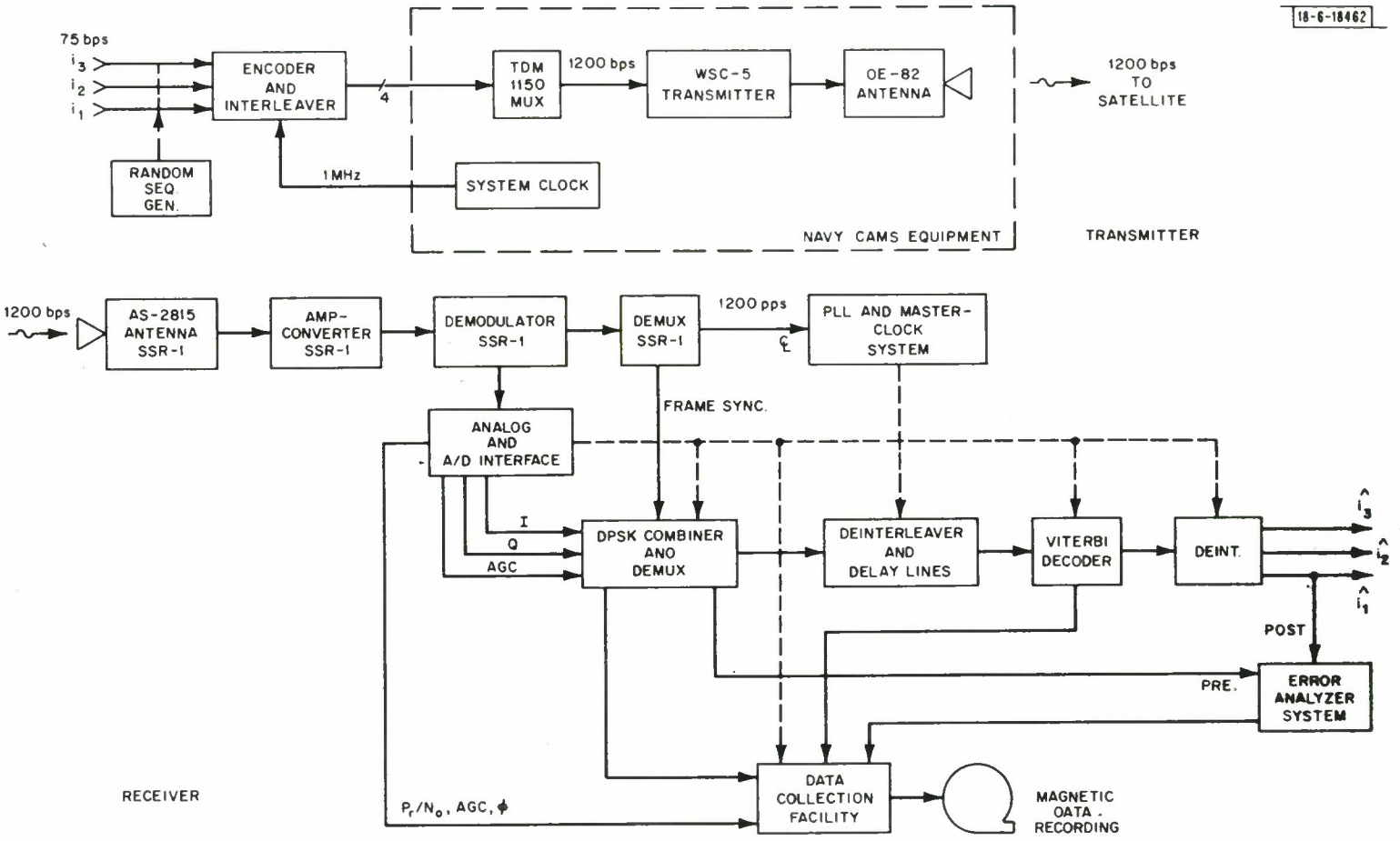


Fig. 3. System Diagram.

locked loop for tracking the carrier with the data stream resulting from the I (in phase) channel output being bit synchronized and matched-filtered. We have added a parallel and identical circuit for the Q (quadrature phase) channel. Moreover 3 bit soft decisions are now being taken from both channels rather than the normal hard quantization on the I channel matched filter output alone. Thus a differentially coherent (in addition to the normal differentially encoded coherent) demodulation scheme and a soft decision capability has been added to the SSR-1. Circuitry has also been added to measure the received power to noise density ratio (P_r/N_o) ratios at the last linear point (as well as P_r alone). The AGC voltage is also sampled and used to place a confidence level on the data; this being done in our DPSK digital combiner network prior to the decoder input. A number of these analog voltages are also digitized for recording.

Lastly, a data collection facility has been implemented with a digital magnetic tape unit as the storage means and a microprocessor used as the controller both for recording and for elementary processing of data. The bulk of the data processing is to be done on an IBM 370 computer facility.

There are a number of important aspects of the experimental program. First and foremost it is intended as a demonstration that time diversity via interleaving and convolutional encoding - Viterbi decoding can provide significant error protection in a fading environment such as that caused by equatorial UHF scintillation. Second, it is important to demonstrate that such a scheme can easily be retrofitted wherever needed to the existing fleet broadcast network. There are of course different options that one might wish to evaluate and the experiment is designed to test some of the

more relevant possibilities. For example using the systematic codes chosen one can protect 1, 2 or 3 information streams at the expense of one overhead parity stream (with of course decreasing error protection). This system would require minimal change in the fleet broadcast operations. Non-systematic codes are also examined mostly to demonstrate the added error protection that such codes give and thus provide some substance for future satellite communications planning where error correcting may be required.

We are also evaluating the performance of the AN/SSR-1 receiver and suggesting modifications that may improve its use in a severe fading environment. For example, the joint use of AGC information and multibit quantization of the matched filter output are obvious means of improving estimates.

There is also great interest in characterizing the UHF scintillation data alone. With a significant period of operation we can collect the fading amplitude and phase statistics and use these for an improved receiver design. For example, rapid phase changes in the received signal may lead to significant dynamic phase error in the SSR-1's Costas phase lock loop and hence to reduced error protection. Accordingly we are recording this dynamic phase error and moreover have designed and will evaluate a demodulation scheme (the so-called differentially coherent combiner) which may minimize the effects of this source of error. Overall the equipment we have constructed should serve as a useful baseline for future designs.

The last section of the report describes the in-house link tests and error rate measurement program. Also included is a summary of the field tests;

however, the complete results will be presented in a follow-on report. We have deferred to this follow-on report all the experimental results and the interpretation of these results in terms of the equipment and operational limitations. That report will also discuss in detail the nature of the scintillation.

II. SSR-1 SYSTEM INTERFACING AND MODIFICATIONS

A. SSR-1 Additions

The AN/SSR-1 is the Navy satellite fleet broadcast receiver which operates at any one of six frequencies between 248.85 and 258.85 MHz. The signal modulation is DPSK at a rate of 1200 bps which is demultiplexed into fifteen 75 bps teletype outputs. The key elements of the receiver are shown in Fig. 4. After frequency down-conversion and amplification the inputs from the four spaced antennas are summed in a regenerative combiner and then demodulated. The combining gain is up to 5 dB above that of a single antenna with the other three antennas terminated in 50 Ω loads. The demultiplexer chassis contains the bit and frame synchronization circuitry along with the demultiplexer and teletype drivers.

An attempt was made to utilize as much of the SSR-1 circuitry as possible so that the new add-on unit would be primarily digital and would not require any significant modifications of the current fleet SSR-1 receiver. Inasmuch as we were investigating different demodulation and decoding techniques a large segment of our add-ons are for experimental purposes only. It is expected that the recommended fleet receiver modification would be very minor and would most probably include only the coherent processing (no quadrature (Q) channel additions) with the ability to perform 3 bit soft decisions on the data (conditioned by an AGC signal). The most extensive addition to the experimental SSR-1 was the inclusion of a quadrature (Q) channel in the PSK demodulator box. The circuitry for this channel duplicates that of the already existing in-phase (I) channel. This demodulation process is more fully described in the next section. Another addition to the Costas loop is a monitor of the I-Q multiplier

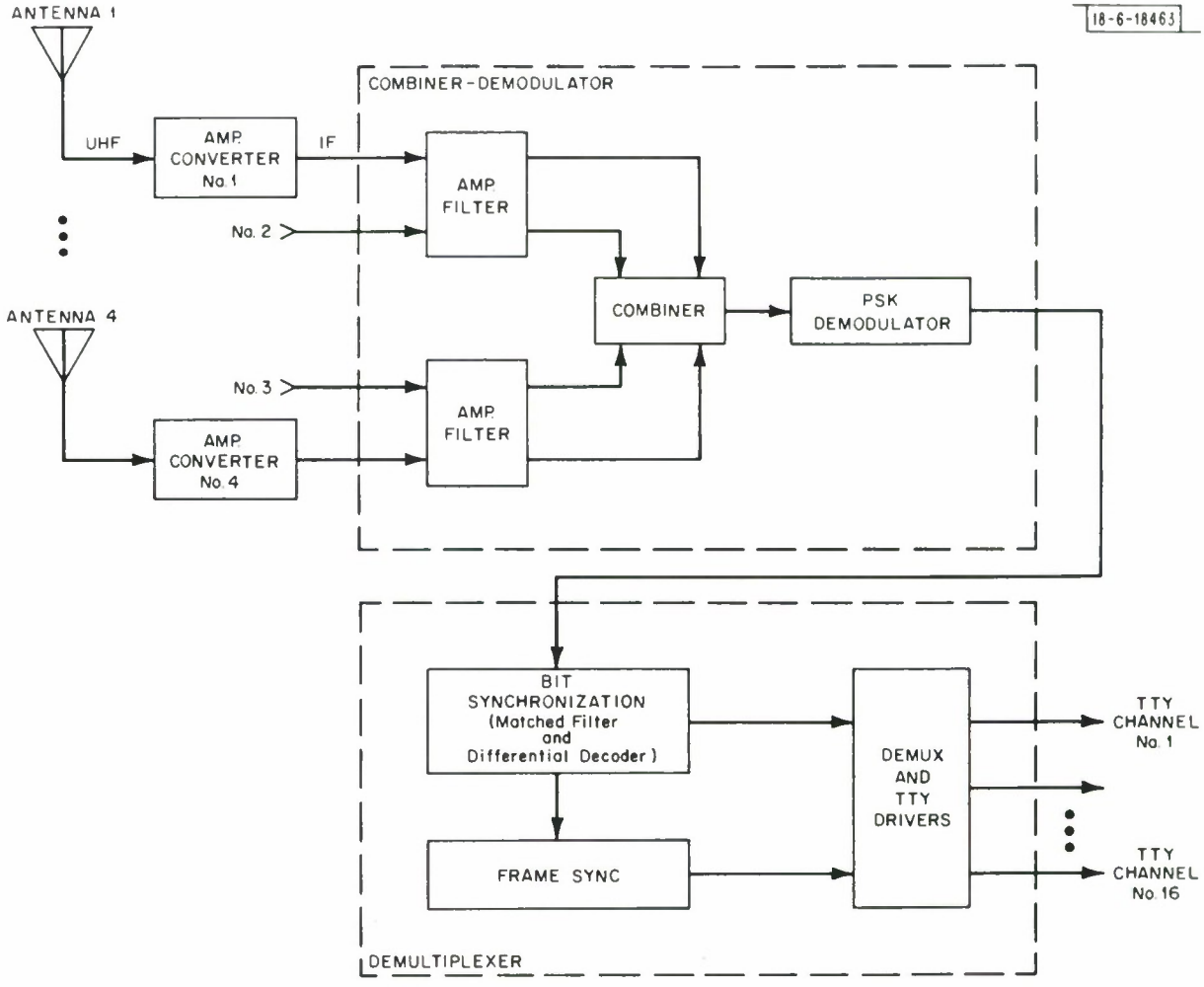


Fig. 4. SSR-1 Block Diagram.

output which is first low pass filtered (f_{co} = 100 Hz) and then digitized to 8 bits. The output is a conditioned dynamic error signal (Lindsay and Simon [5]) which in turn is related to the phase error. (For the situation when the signal energy is much greater than noise the dynamic error signal ε is approximately proportional to ϕ (τ), the phase error of the loop.) This signal is sampled 150 times during a 2 sec period at a sample rate of 1200 Hz. These samples are recorded on magnetic tape for later spectral analysis.

The SSR-1 "blanker" signal is monitored and after a hard decision is recorded at a 75 Hz rate. The signal is at a logic "one" whenever the receiver "blanks" due to excessive input signals. There are two other outputs which are needed for the proper operation of the add-on digital section. The master timing system is slaved to a 1200 Hz clock which comes from the bit synchronizer portion of the SSR-1. From the frame synchronization section of the SSR-1 demultiplexer the 75 Hz frame sync stream is picked off and is used to initialize our version of a demultiplexer.

Since signal level estimates are needed both for the error rate curves as well as the scintillation characterization it was necessary to monitor the IF (19.95 MHz) signal at a point in the receiver prior to the AGC. (The signal was taken off test point 5 in the amplifier-filter section of the SSR-1 receiver.) Figure 5 is a block diagram of the circuit which measures both signal and signal to noise ratio. The outputs are digitized (8 bits) and sampled at 50 Hz for eventual digital recording. Both of these outputs are calibrated periodically by injecting (at RF) a precisely measured signal and recording the A/D output. A complete characteristic is obtained by inserting a precision RF attenuator in the line and using 1 dB attenuation steps until

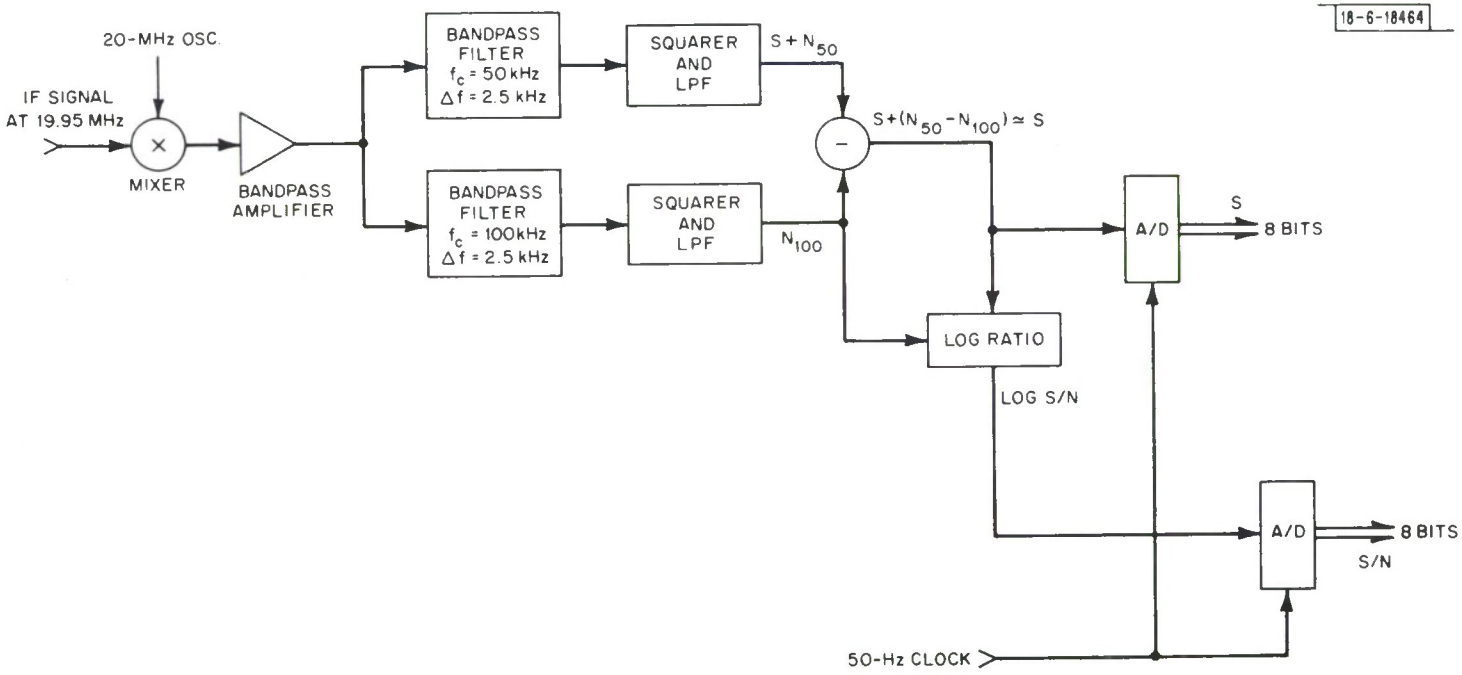


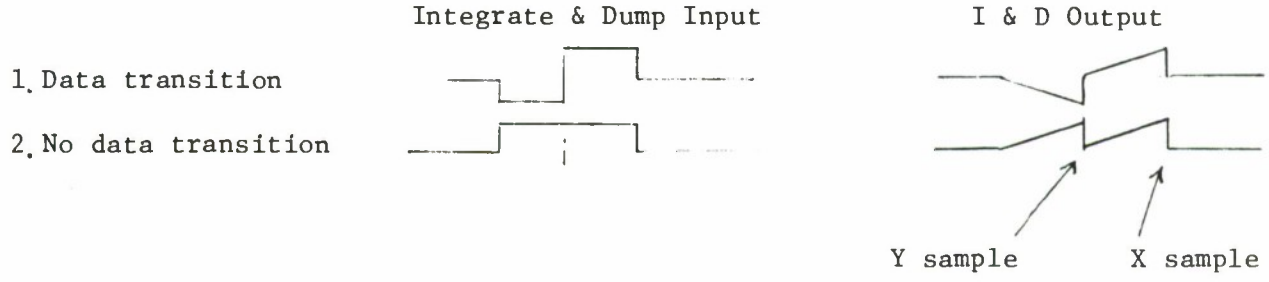
Fig. 5. Signal and Signal-to-Noise Measurement Circuit.

the expected received signal range is spanned. It should be observed that these outputs are representative of the true values of S and S/N only if the noise spectrum is flat (over 52.5 kHz). As a backup the noise spectrum is routinely checked and the S and S/N ratio is measured directly with an HP spectrum analyzer.

B. DPSK Demodulation

From Fig. 6 it is seen that the in phase (I) and quadrature (Q) channel signals used to develop the chip streams originate at the output of the low pass filters in the SSR-1's Costas loop. In the normal SSR-1 configuration the I signal alone feeds a matched filter in the bit synchronizer and its output is hard quantized to 0 or 1. We have added circuitry to level shift, amplify and digitize (to 3 bits) the matched filter output. We have also added a parallel and identical channel for the Q signal (including assorted filters, matched filter and a 3 bit A/D converter) so that we can perform soft decision differentially coherent combining in addition to the usual coherent combining. The combining and DPSK decoding circuitry is in a digital drawer along with the digital system timing and deinterleaver complex.

The matched filters in the bit synchronizer section are integrate and dump circuits matched to a single chip. For a DPSK signal the appropriate matched filter would be matched to 2 consecutive chips. Moreover since polarity is irrelevant in DPSK we need only match against the two waveforms shown.



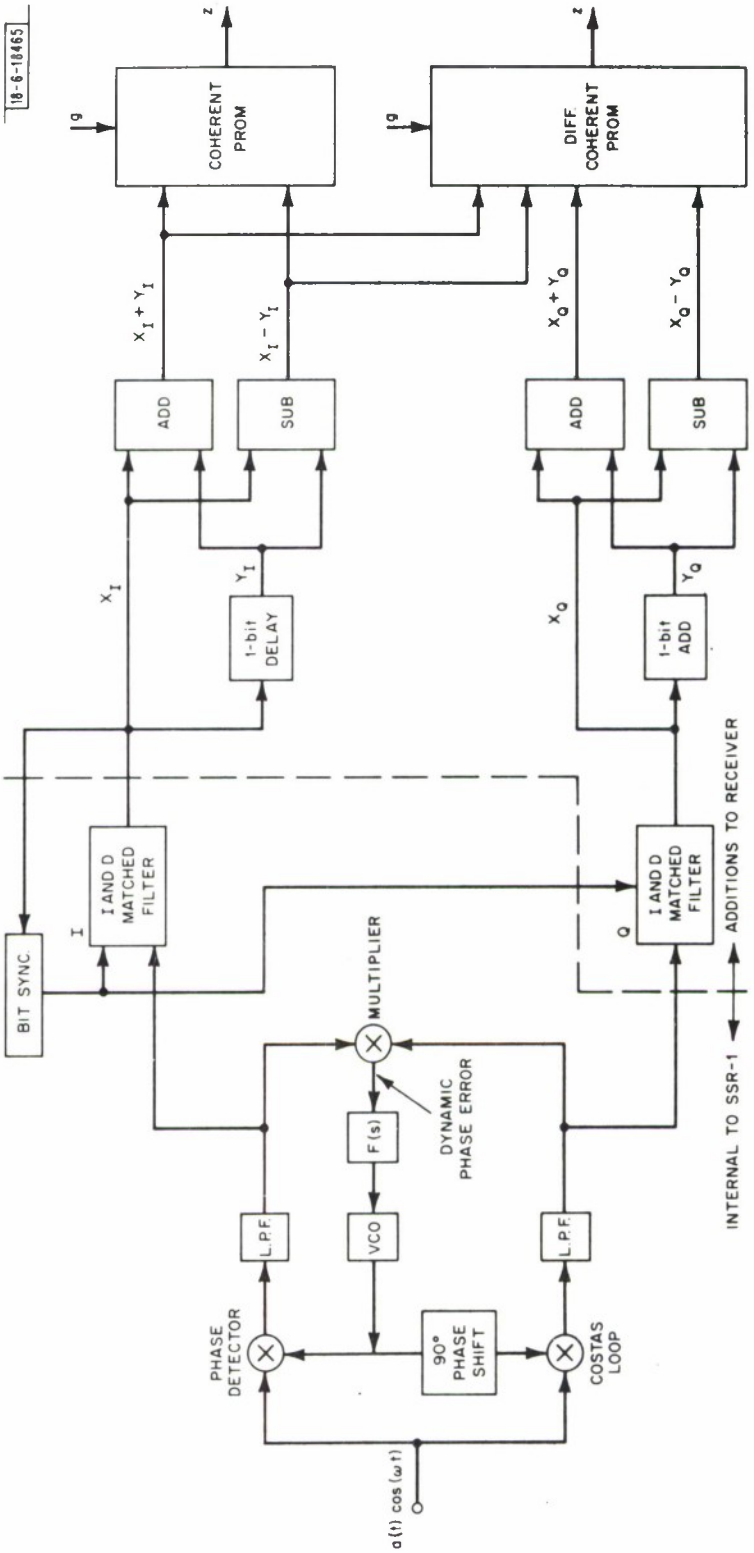


Fig. 6. Coherent and Differentially Coherent Demodulator.

Let the present samples for the in phase and out of phase components be denoted as X_I, X_Q and the previous (one bit delayed) samples as Y_I, Y_Q. The appropriate maximum likelihood receiver (which also includes the DPSK decoding) can be realized with a few 4 bit address PROMS and 4 bit registers as shown in Fig. 6. The DPSK receiver would make a 'soft' decision between two functions which are themselves functions of $|X+Y|$ and $|X-Y|$ (the absolute values reflecting the irrelevance of DPSK polarity). If we are operating in the coherent mode (using the in phase (I) channel only) then a very good approximation for the maximum likelihood receiver (for strong signals) would be a circuit that provides the following output

$$Z = g\,[\,|X_I + Y_I| - |X_I - Y_I|\,] \qquad (1)$$

where g is a scale factor that compensates for the AGC behavior. Z, X, Y are all 3 bit 2's complement binary numbers.

If the phase remains almost coherent over two chips even though the dynamic phase error in the Costas loop is considerable, then we want to utilize the information in the quadrature (Q) channel. This is termed the <u>differentially coherent</u> case. By analogy with the maximum likelihood receiver for incoherent combining, the appropriate output is then approximated by

$$Z = g^2\,[\{|X_I + Y_I|^2 + |X_Q + Y_Q|^2\} - \{|X_1 - Y_1|^2 + |X_Q - Y_Q|^2\}] \qquad (2)$$

Two PROMS were programmed to perform the calculations required by equations 1 and 2.

C. System A.G.C.

The SSR-1 receiver has an AGC loop whose function is to provide a fairly constant signal for the tracking loops. It is reasonable then to want to assign a higher probability of bit error when the AGC is at its maximum and the received symbol is small (in the SSR-1 at the maximum negative voltage). In particular for those soft decisions during times of a deep fade we want to effectively assign a lower matched filter output voltage. As a result of the simulations the AGC will be utilized in the following way. The AGC is passed through a bandpass filter with 3 dB points at approximately .01 and 3 Hz. The output of the filter is digitized to 2 levels. The binary count 11 is assigned to a range of voltages from a little below 0V to the upper limit of +2V. For the voltage range from a little below 0V to -2V the region is equidivided into three regions corresponding to 10, 01 and 00. This filtered and digitized AGC voltage is used as address elements for the combiner ROMs which in turn put out the demodulated 3-bit signal estimates. The effect of this is to assign a strong signal estimate when the incoming signal level is either constant or at a local peak. When the signal suffers a strong fade with a fade duration between a few tenths of a second to a few seconds, the corresponding bits will be assigned a "lesser" signal estimate.

There is also another AGC signal which results from picking off the same SSR-1 AGC loop voltage and then passing it through a 3 Hz low pass filter and an 8 bit A/D converter. The sampling rate is 25 Hz and all samples are recorded on magnetic tape for future processing.

III. DIGITAL SYSTEM TIMING

A. Master Digital Clocks

We have had to develop system clocks both for the digital "transmitter" and "receiver" boards which are compatible with the CAMS (Communications Area Master Station) transmitter system and the SSR-1 receiver system, respectively. The clock pulses needed by the "transmitter" section come from the CAMS clock distribution system. We require a clock from the CAMS of n.1200 pps where n is a fairly large non-prime integer. All our transmitter clocks result from dividing down from this frequency (1.2288 MHz).

The receiver clocks are generated from a 1200 pps train produced in the SSR-1. A digital phase locked loop with a maximum frequency of 600 kHz is followed by a digital divide-down chain of 500 which provides all the necessary timing signals. Standard 4 lines to 10 TTL decoder chips are utilized to give 10 phase clocks. A further dividing down from 1200 pps to 1/2 pps gives the the A/D strobes and the corresponding interrupt pulses required by the data recording facility. The digital phase lock loop is discussed more fully in Appendix B.

B. Interleaving Strategy for Systematic Codes

The following argument can best be followed by reference to Fig. 7 (and Figs. 9 and 10 for more detail). Up to three parallel information streams are generated and transmitted without any manipulation and therefore can be 'read' by any unmodified SSR-1 receiver system (which does not use the parity check channel). Strong scintillation would of course cause blocks of erroneous data resulting in a relatively high error rate. The inclusion of an encoder and

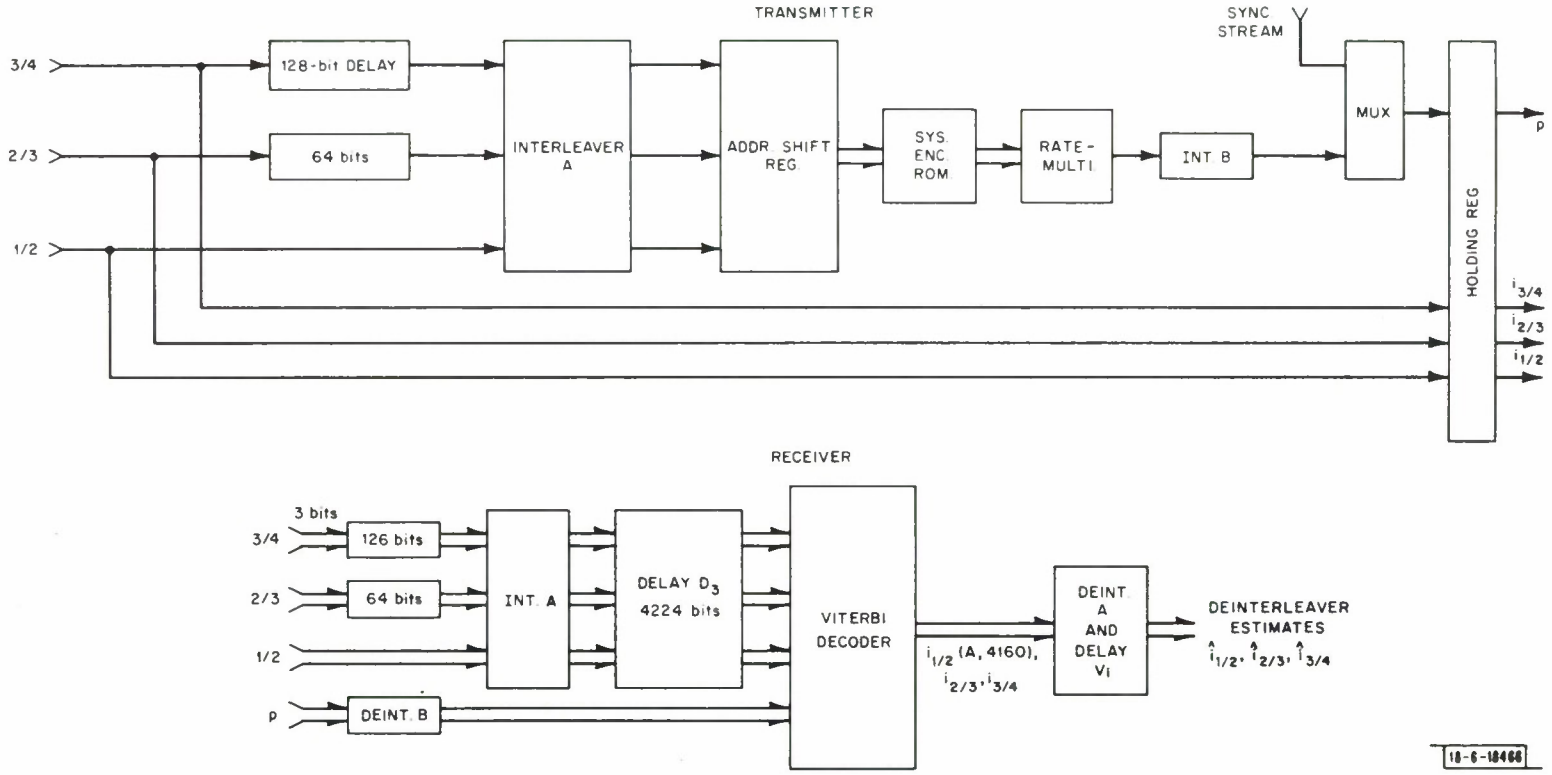


Fig. 7. Interleaver Configuration for Systematic Codes.

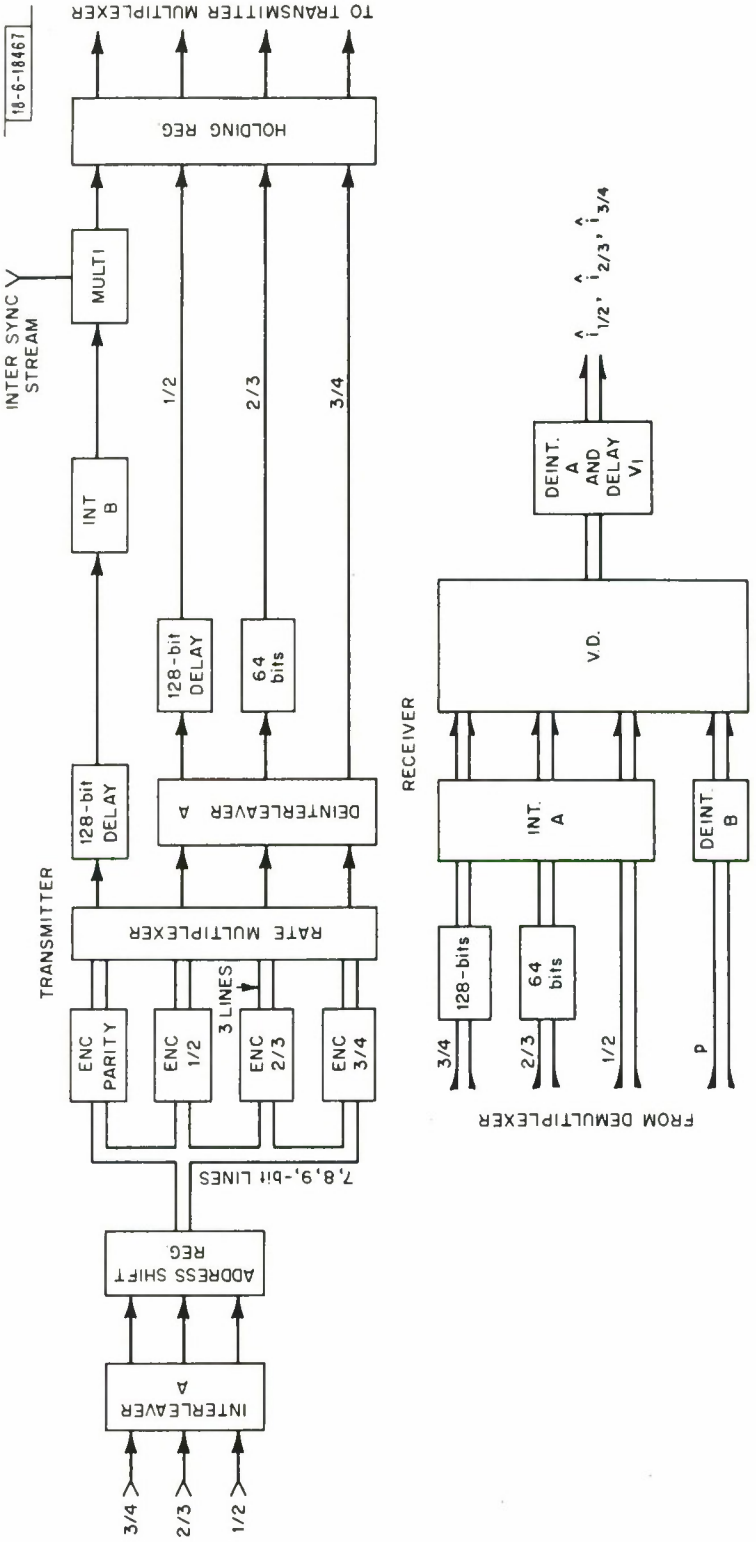


Fig. 8. Interleaver Configuration for Non-systematic Codes.

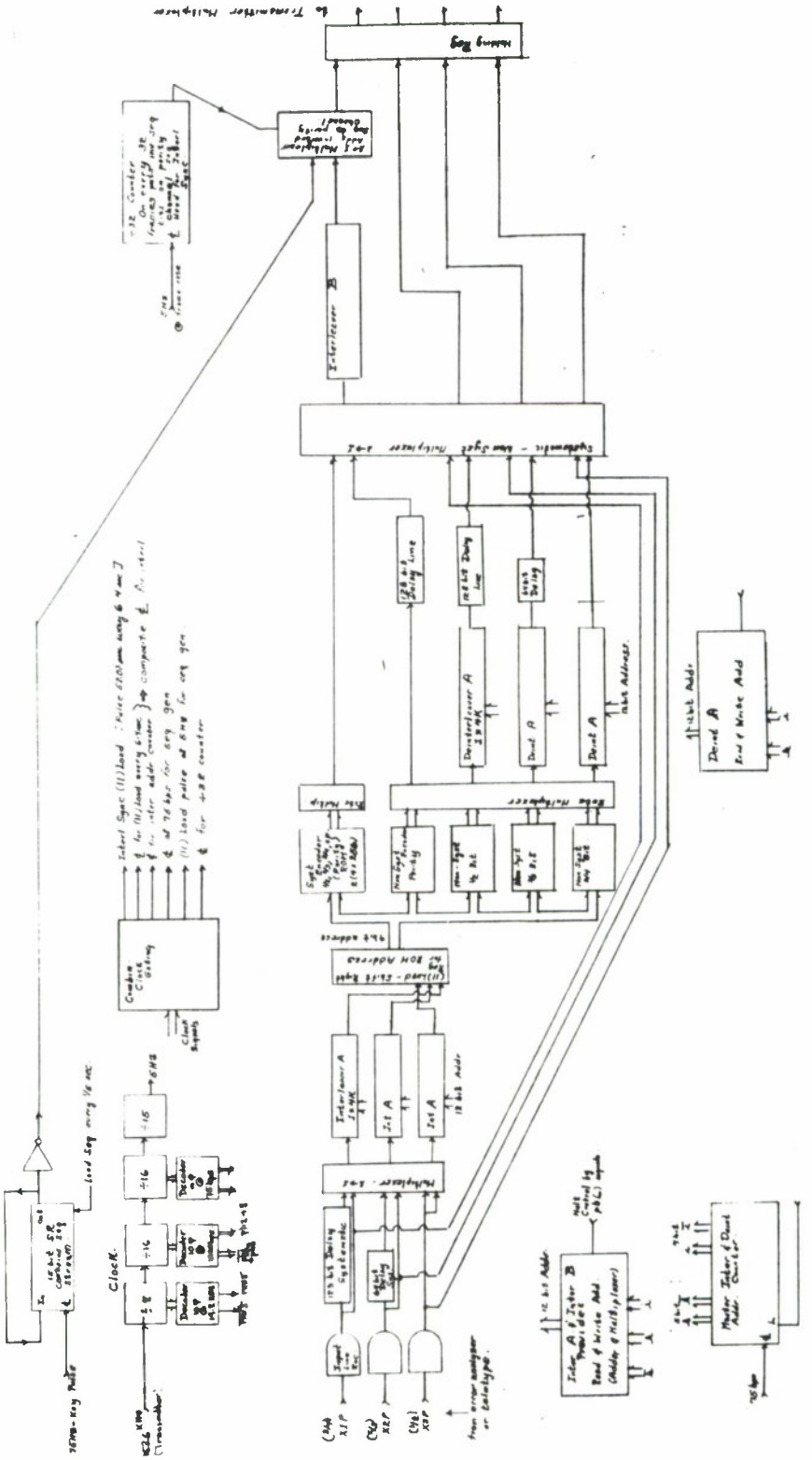


Fig. 9. Transmitter Functional Diagram.

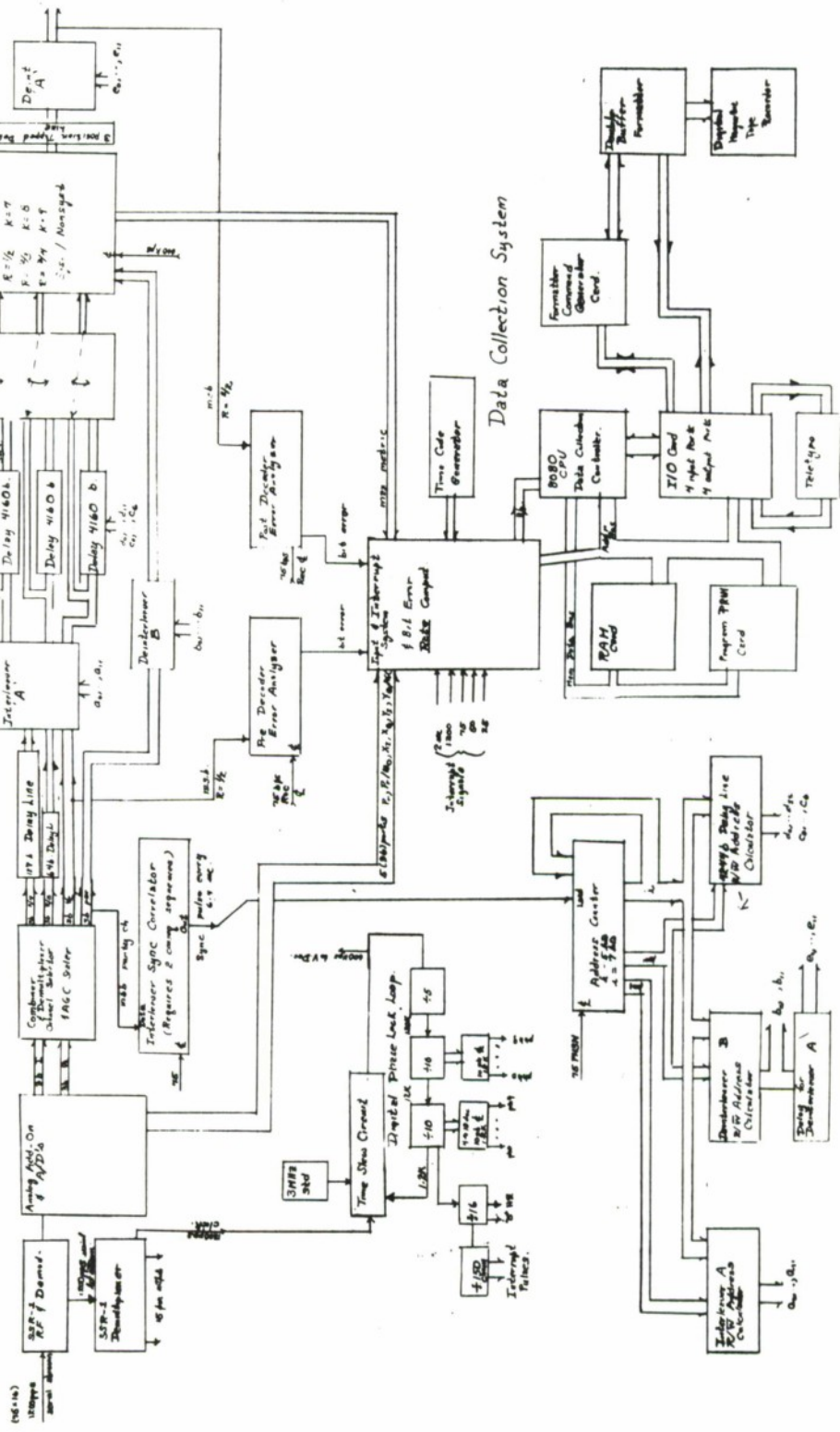


Fig. 10. Receiver Functional Diagram.

hence an additional (parity check) channel would provide a degree of redundancy and therefore provide some error correcting capability. However, if a large number of errors occur grouped together most practical decoders will result in an almost negligible improvement in error rates. Viterbi decoders work best when the error locations are randomized. The randomization is accomplished with the use of interleaving (and to a lesser extent with the 64 and 128 bit delay lines). Observe that the original transmitted bits (which we designate by $i_{1/2}$, $i_{2/3}$, $i_{3/4}$, p) are received with blocks of errors resulting from the scintillation fading. These streams are then interleaved in the receiver (by interleaver A) and thus the Viterbi decoder will see only data streams with randomized error locations. The 128 bit and 64 bit delay lines preceding interleaver A reduce the probability of having simultaneous errors on the three channels. We designate the three data streams presented to the Viterbi decoder (ignoring for the present the meaning of the 4160 bit delay lines) as $i_{1/2}$ (A, 4160), $i_{2/3}$ (64, A, 4160) and $i_{3/4}$ (128, A, 4160). It follows that the transmitter encoder must see the same relative sequence and this is evident from the transmitter diagram. The encoder produces a parity check stream which is then interleaved by interleaver B and transmitted. This transmitted stream suffers transmission errors in bursts and on reception the deinterleaver B results in the original parity check stream with randomized errors. This parity check stream is then presented to the Viterbi decoder. It should be observed, however, that the process of interleaving and deinterleaving (B) causes a relative channel delay of 4160 bits (due to the particular interleaver design selected) which was compensated for in the delay lines inserted in the information bit lines as noted above.

The Viterbi decoder outputs are designated as $\hat{i}_{1/2}$ (A,4160), $\hat{i}_{2/3}$ (64, A, 4160) and $\hat{i}_{3/4}$ (128, A, 4160) and are still interleaved. There is also a delay $(V)_i$ inherent in the Viterbi decoder which is related to the length of the path memory as well as a delay associated with the time required for the information bits to be shifted through the encoder. This resultant delay varies for the three different code rates. However, we only need to add delays less than 32 bits in any case because again we only need to satisfy row synchronization as discussed in the section on interleaver synchronization. Finally we require a deinterleaver (A) to convert the output streams into estimates of the original text. However, it should be noted that these estimates will have a delay of $(2(4160) + V_i + L_i)/75$ sec relative to the original text (L_i is 128 for 3/4 rate, 64 for 2/3 rate and 0 for the 1/2 rate and V_i is 22, 17 and 2 bits, respectively).

C. Interleaving Strategy for Non-Systematic Codes

The non-systematic codes are not readable by the unmodified SSR-1 receivers since all the transmitted bits are encoded as parity check bits (see Fig. 8). As in the systematic case the receiver's interleaver A and the 128 and 64 bit delay lines cause the randomization of errors. The outputs from the receiver's interleaver (A) are directly presented to the Viterbi decoder (the 4160 bit delay lines being bypassed). At the transmitter the input data $i_{1/2}$, $i_{2/3}$, $i_{3/4}$ is applied directly to the interleaver (A) before being encoded in the (up to) four non-systematic encoders. After the encoders, three of the channels feed deinterleaver A and the fourth channel feeds interleaver B. At the receiver, the three channels are fed into interleaver A and the fourth into deinterleaver B. The net result of these complementary operations is to introduce a fixed 4160

bit delay on all four channels. It is observed also that the 128 and 64 bit delay lines on both the transmitter and receiver are configured to give a net 128 bit delay on all 4 lines as well. As a result, the input bit streams to the Viterbi decoder are the same as the bit streams emanating from the encoders with, however, a fixed 128 + 4160 = 4288 bit delay. There are of course differences during the period of time when the interleaver sync code is multiplexed on the channel containing interleaver B (and of course, difference due to the presence of bit errors resulting from noise or fading).

As in the systematic case there is a large throughput delay of $[2(4160) + 128 + V_i]/75$ sec. Again, this is one of the penalities of a time diversity error protection scheme.

D. Interleaver Synchronization (Row Synchronization)

It is necessary to provide synchronization of the interleavers in the transmitters with those deinterleavers and delay lines in the receiver. It is shown in Appendix D that the interleavers, deinterleavers and delays can each be regarded as stacks of delay lines (in our case the stack height being 32). If a particular data bit is first injected into the jth row of the first interleaver, it must then remain in the jth row of all future interleavers, deinterleavers and delay lines. Interleaver synchronization then refers to rows only and if in the transmitter the particular bit is assigned to a row j, then in the receiver the same bit must be assigned to the jth row as well. Towards this end a synchronization code of 15 bits is transmitted on a selected parit channel (in place of the normal parity bits). This group of 15 bits is transmitted every 6.4 sec and is not interleaved. It should be noted that this

causes about a 1/2 $(\frac{15}{(75)(6.4)})$ = 1/64 erasure rate on this channel. (This is, however, partially offset by the fact that these sync bits are weighted as little as possible when computing the soft decisions metrics). The receiver correlates the hard decision bit stream from the selected parity channel with the known stored synchronization code and gives an output only when all consecutive 15 bits match. To provide further false sync protection, we require perfect matching of two consecutive groups of sync pulses separated by the normal 6.4 sec. The probability of having a false synchronization at any time is reduced to no more than $(.5)^{-30} = 1.25 \times 10^{-9}$. After synchronization is established, there may be intense scintillation periods where no further sync pulses are generated. However the sync mechanism will "fly-wheel" through this period. In addition during periods of little or no scintillation, a false synchronization will not last longer than 2(6.4) = 12.8 sec.

The particular sync code we use is the inverse of the 15 bit code use to establish frame sync for the SSR-1 receiver. It would be ideal if one could put this inverted stream on the frame sync channel. This would not affect the SSR-1 from finding frame sync at all and would have a number of beneficial effects. It would eliminate the errors in the selected parity channel and hence lower the overall P_E versus E_b/N_o curves. The probability of false synchronization would also diminish significantly and, moreover, the overall sync hardware would be less.

E. Encoder-Decoder Synchronization (Column Synchronization)

At the encoder (or encoders for the non-systematic codes) we have from 1 to 3 input bits depending on the code rate and 2 to 4 output chips are generated.

For the systematic codes only one parity check bit is generated (the other output bits being the input bits), whereas for the non-systematic codes all the output bits are parity check bits. It is essential that the same time relationship (no relative bit delays between channels) holds at the input to the decoder. If there are relative bit delays the final output will be totally unrecognizable. Column synchronization then refers to the inclusion of special delay lines into the design which provide this proper time relationship between the 2 to 4 channels. The column synchronization turns out to require a significant number of chips in the overall system.

The most prominent example of this is the 4160 bit delay lines that need to be added to three of the channels in the systematic configuration in order to compensate for the delay introduced by the interleaver (B)-deinterleaver (B) combination. In the non-systematic rates the transmitter's 128 and 64 bit delay lines also serve for column synchronization.

It is an article of faith that a differential channel delay will not be introduced by the CAMS multiplex system over which we have no direct control. One must realize that if this were so one would have to introduce sync codes in each and every channel, thereby significantly increasing the complexity of the receiver digital portion.

IV. CODING SCHEME

A. Convolutional Encoder

The convolutional codes have been extensively studied (i.e., Viterbi [4]) and our summary is meant only to clarify some of the hardware issues. The general binary input - binary output encoder consists of a K-stage shift register and n modulo-2 adders. Each of the modulo-2 adders is connected in a specified pattern to certain of the shift registers (see Fig. 12). These connections (also called generators) determine the code. Information bits are shifted into the encoder k bits at a time and after the shift the n mod-2 adder outputs are sampled sequentially to give the channel symbols. Note that n channel symbols are generated for every k input bits resulting in a code ratio $R = \frac{k}{n}$. In our experiment we are investigating only the code rates 1/2, 2/3 and 3/4.

There is a special class of codes, the systematic codes, where k of the n mod-2 adders consist of a single special connection such that the information bits themselves are a subset of the n channel symbols. More generally the non-systematic codes, where all the mod-2 adders have more than a single input, have better error correcting capability mainly because each output channel symbols depends on a combination of nearby input information bits. It should be obvious that we want to have <u>each</u> output channel symbol depend on a number of neighboring information bits, where the number of bits is subject, however, to minimum distance considerations. In this way loss of a single or even a few near-contiguous output channel symbols does not result in a corresponding highly probable error in the decoded output information stream. It is also intuitive that the error correcting ability should increase with increasing register length K. It should be noted that any particular information bit is in the encoder for K/k cycles.

Bucher has assembled either from a literature search or by a computer search optimum or near optimum convolutional codes which we have used with our Viterbi decoder. These are listed along with the code generators $g_1 g_2 g_3 \cdots g_k$.

TABLE I
CODE GENERATORS

R	K	$(g_1 g_2 \cdots . g_K)$	free distance d_f	Source
		Systematic		
1/2	7	1 1 1 1 0 1 1	6	Bucher
2/3	8	1 1 0 1 1 0 1 1	4	
3/4	9	1 1 1 0 1 1 1 0 1		
		Non-Systematic		
1/2	7	1 1 1 1 0 0 1	10	Oldenwalder [7]
		1 0 1 1 0 1 1		
2/3	8	1 0 1 1 0 1 1 0	8	Paaske [8]
		0 1 1 1 1 0 0 1		
		1 1 1 1 0 1 1 1		
3/4	9	1 0 0 1 1 1 0 1 0	6	Paaske
		0 1 0 0 0 1 1 0 1		
		0 0 1 0 0 1 0 1 1		
		1 1 1 1 0 1 0 0		

In terms of hardware realization the n mod 2 adders are programmed on read only memories (ROM's) which are in turn addressed by the K-stage shift register. We are also investigating the effects of coherent versus differentially coherent detection on the error rate. All the possibilities are shown in Table II.

TABLE II

CONVOLUTIONAL CODES

	Coherent	Differentially Coherent
Systematic Code	R=1/2 2/3 3/4	1/2 2/3 3/4
Nonsystematic Code	1/2 2/3 3/4	1/2 2/3 3/4

B. Viterbi Decoder

It is well recognized that the Viterbi decoding algorithm yields maximum likelihood decisions on data that has been first convolutionally encoded and passed through a memoryless channel. In our case the long duration fades imply memory but judicious use of interleaving yields an essentially memoryless channel. As a result the channel can be adequately described as the classical memoryless binary symmetric channel with additive white Gaussian receiver noise plus a signal amplitude distribution which is close to Rayleigh as a consequence of scintillation.

Let us assume we have an input data stream $\underset{\rightarrow}{z}$ where each element of this vector represents 1, 2 or 3 information bits (corresponding to the number of parallel and independent fleet broadcast channels one wishes to protect). The elements of this vector are interleaved (resulting in $\underset{\rightarrow}{z}_I$) and presented to either a single convolutional encoder (output $\underset{\rightarrow}{u}_s$) in the systematic case or a bank of encoders in the nonsystematic case (output $\underset{\rightarrow}{u}_{ns}$). In either case we represent the channel input to be the stream $\underset{\rightarrow}{x}$ where each element of $\underset{\rightarrow}{x}$ represent 2, 3 or 4 channel symbols. In the systematic case $\underset{\rightarrow}{x}$ is equivalent to the concatenation of $\underset{\rightarrow}{z},\underset{\rightarrow}{u}_s$ and in the nonsystematic case $\underset{\rightarrow}{x}$ is equivalent to $\underset{\rightarrow}{u}_{ns}$. The transmitter's deinterleavers A and interleaver B and the receiver's deinterleaver B, interleaver A and delay line can be considered as part of the conceptual channel. The only effect of these interleavers and delay lines is to introduce a channel delay. In addition, the DPSK encoder, modulator, demodulator as well as the specially constructed incoherent or coherent combiner with DPSK decoder are part of the channel.

Similarly the final deinterleaver A is considered part of the decoder. The received stream presented to the decoder is represented by $\underset{\sim}{y}$ and the final output stream from the decoder by $\hat{\underset{\sim}{z}}$ where ($\hat{}$) signifies the estimate of the original input stream. It should be realized that $\underset{\sim}{y}$ differs from $\underset{\sim}{x}$ due to the effects of receiver noise and signal fading. A maximum likelihood decoder is one which first investigates the conditional probability functions $p(\underset{\sim}{y}|\underset{\sim}{x}^{(i)})$, where $\underset{\sim}{x}^{i}$ is one of the possible transmitted sequences, and then selects that sequence with the maximum probability. Now if the random process is stationary and if the interleaved channel is effectively memoryless and hard decisions are used, then the probability of making a channel symbol error is just p. The probability of having received a particular sequence that differs in exactly d_i places is given by (for N received symbols)

$$p(\underset{\sim}{y}|\underset{\sim}{x}^{i}) = p^{d_i}(1-p)^{N-d_i} \tag{3}$$

Taking the log gives the function termed the log likelihood.

$$\log p(\underset{\sim}{y}|\underset{\sim}{x}^{(i)}) = -d_i \log\left(\frac{1-p}{p}\right) + N\log(1-p) \tag{4}$$

For $p < \frac{1}{2}$ maximizing the probability function (or equivalently the log of the probability) is identical to finding the test sequence or path that differs by the fewest channel symbols from the received sequence (i.e., Hamming distance is smallest).

If, however, we are able to quantize the output to the matched filters to a number of bits greater than 1, then it seems intuitively obvious that one

should be able to take advantage of the additional information provided that the quantization levels are chosen appropriately. As long as the channel is memoryless we can write

$$p(\underset{\sim}{y}|\underset{\sim}{x}^{(i)}) = \prod_{k=1}^{N} p(y_k|x_k^{(i)}) \tag{5}$$

where N is the total number of channel symbols transmitted. The received signal out of the matched filter and combiner in the absence of AGC compensation could be represented by the symetrical bipolar signal y_k where

$$y_k = (a_s)_k \cdot x_k + n_k$$

We have assumed that the value of x_k is either ± 1. Likewise, $(a_s)_k$ is a random variable whose amplitude prior to the demodulator was Rayleigh distributed and n_k is a white Gaussian noise process with independent samples. In this case

$$p(y_k|x_k^{(i)}) = \frac{1}{\sqrt{\pi N_o}} e^{\frac{-(y_k - (a_s)_k x_k)^2}{N_o}} \tag{6}$$

Therefore the log likelihood is given by

$$\log p(\underset{\sim}{y}|\underset{\sim}{x}^{(i)}) = C_1 \sum_{k=1}^{N} (a_s)_k y_k \cdot x_k^{(i)} + C_2$$

where C_1 and C_2 are independent of the tested path i. The decision rule would

be to choose that sequence $x^{(i)}$ which maximizes $\sum_{k=1}^{N}(a_s)_k y_k \cdot x_k^{(i)}$. Use of this decision rule would required knowledge of $(a_s)_k$ which up to this point is an unknown variable. The AGC voltage is applied to the combiner PROM as shown in Fig. 6 (signal g). If the AGC voltage is represented by a_g, then the desired effect of the AGC is to make the product $(a_g)_k(a_s)_k$ more nearly a constant value as compared to $(a_s)_k$ alone. In this case the admittedly heuristic decision rule is to find the test sequence that maximizes

$$\sum_{k=1}^{N} (a_g)_k(a_s)_k\, y_k \cdot x_k^{(i)} \simeq C_3 \sum_{k=1}^{N} y_k \cdot x_k^{(i)} \tag{7}$$

In practice y_k is quantized to 2 bits and a sign bit and a_g to 2 bits. The quantization step sizes were optimized from simulation studies which use a fading distribution suggested by actual recorded scintillation data.

It should be noted that our realization of the Viterbi decoder operates on up to four parallel input channels at the same time. It is then more convenient to write the inner product metric in a different way. Let $\tilde{y}$ denote a m by N/m matrix of channel output levels and $\tilde{x}$ a N/m by m matrix (m is the number of input channels and N the total number of input chips of message). The appropriate decision rule would be to find that path out of all possible paths that maximizes the metric

$$\max_i \sum_{k=1}^{N/m} \left(\sum_{j=1}^{m} y_{kj} \cdot x_{jk}^{(i)}\right) = \max_i \sum_{k=1}^{N/m} m_k^{(i)} \tag{8}$$

Each term within the inner parentheses is termed an incremental (or branch) metric. In the decoder implementation we have used a mapping such that if the received signal $|y_{kj}|$ is at its largest value and x_{jk} agrees with y_{kj} in sign then a value of +7 is assigned to the product $x_{jk} \cdot y_{kj}$. If on the other hand $|y_{kj}|$ is again at its largest but x_{jk} and y_{kj} differ in sign then a value of 0 is assigned to the product. The intermediate values fall in between. The y_{kj} values are given in 2's complement form with positive values representing a logic '1' and negative values a logic '0'. Likewise x_{jk} has only logic '1' or '0' and the mapping circuit is shown in Fig. 11. The branch metrics are all positive and hence the accumulated metric is a monotonic function of N.

The total number of paths one would have to keep track of grows as 2^N unless some means are developed which continually discard paths. The Viterbi algorithm is one such means. Traditionally it has proved useful to describe the input-output relations of the convolutional encoder either by a directed state diagram or by an equivalent tree or trellis diagram (Fig. 12). For the directed state diagram the K bit shift register stages can be divided into k input bits (k bits are shifted in at a time) and the K-k state bits. Therefore there are 2^{K-k} different states possible (or 2^{K-k} nodes) and 2^k different input combinations (or 2^k branches emanating from each mode). Each branch is labelled both with the input bits as well as with the convolutional encoder output bits. Then, given an initial state and an input data stream (or parallel input streams), we can follow the progress from state to state as successive input bits are brought into the encoder. We can more easily follow our progress on the equivalent lattice diagram where every possible path is

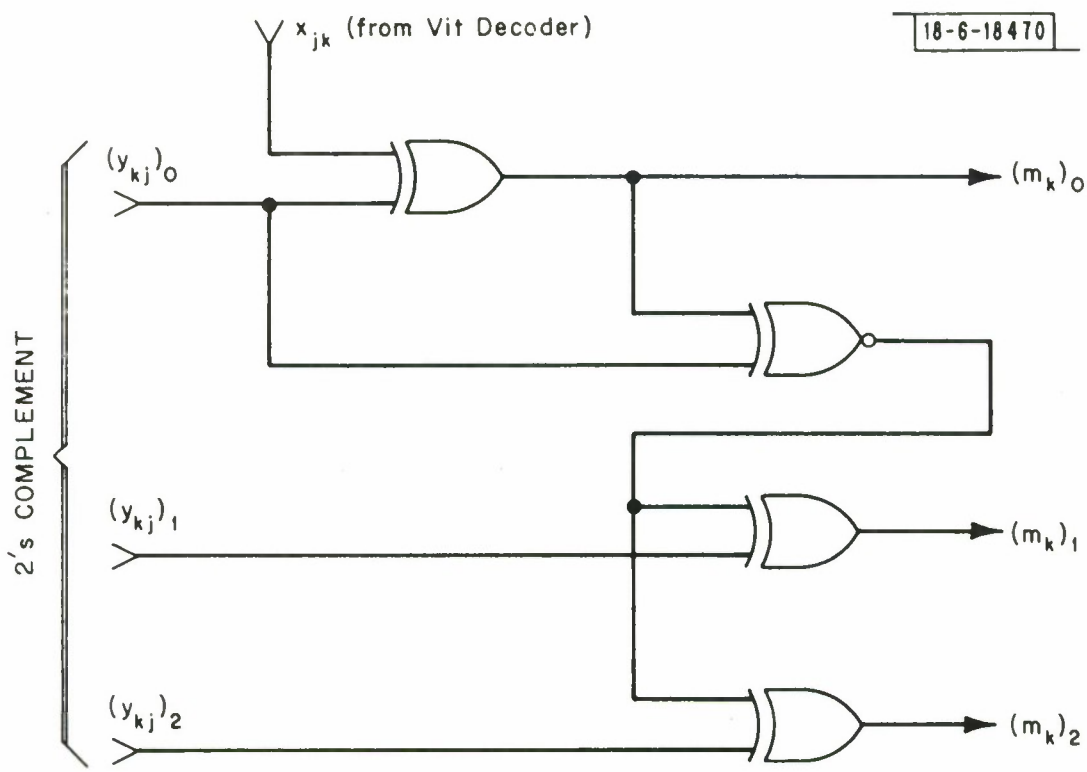


Fig. 11. Incremental Metric Map.

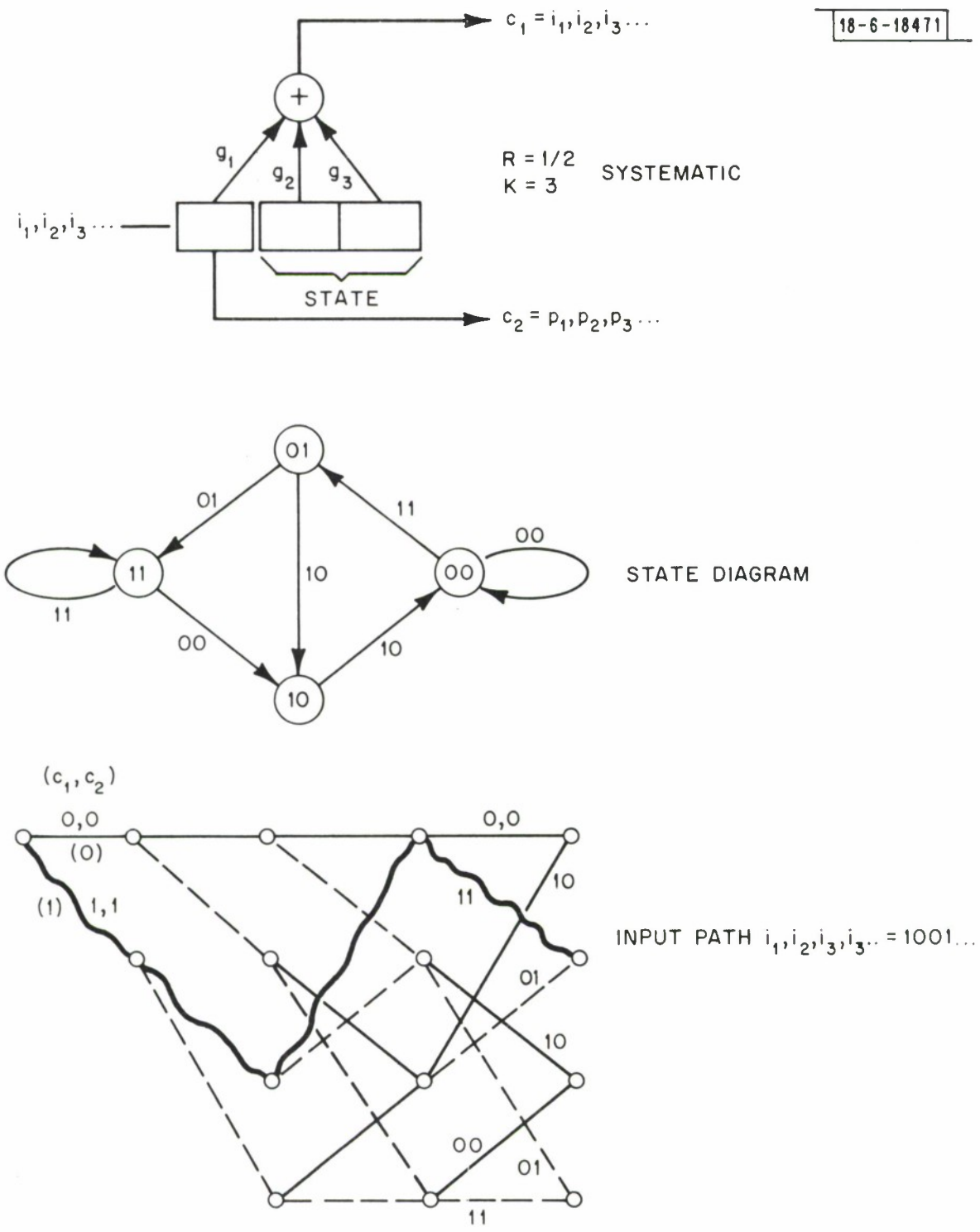


Fig. 12. Convolutional Encoder, State Diagram and Lattice.

represented and can in fact be traced. The Viterbi algorithm can be simply described with the aid of this lattice. Note that the nodes of the lattice represent the encoder state and a vertical column of nodes represents all the 2^{K-k} possible states. The 2^k branches into the nodes again represent the number of possible input bit combinations that led to this state. Let us assume that after receiving the i group of input bits we are at the ith node column. It will be stated (without proof) that there at this time 2^K possible 'paths' whose accumulated metrics have been computed and stored. For <u>each</u> state we examine the 2^k paths coming into the state and only retain that path which has the largest accumulated metric. After performing this task for all the states in the column we are left again with 2^{K-k} saved 'paths' as well as 2^{K-k} associated accumulated metrics. Now as we input the next group of bits we again have $(2^{K-k}) \cdot 2^k = 2^K$ paths to examine at column (i+1) and again we retain those 2^{K-k} paths with the largest metrics. The important point is that number of computations grows as the constraint length K and not as the message length N. The paths saved are no more than selected subsets of the set of all possible input sequences up to the ith input bit. Clearly saving the subset of paths with maximum accumulated metric is compatible with the decision rule of Eq. (8). The problem remains to determine which of the remaining 2^{K-k} paths is best. It turns out, however, that if one compares all the 2^{K-k} paths and looks at the <u>earlier</u> input bits, one will find that they are nearly identical (all the other paths were discarded). If we go back far enough, then, we can select the input bit on any one of the saved paths. Simulation has shown in fact that if one chooses that path out of the 2^{K-k} remaining with the current

maximum metric and take as output those input bits a few constraint lengths earlier in bit times, then the probability of error is very small. This also allows us to retain a finite path length and hence allows a modest size RAM. Another hardware consideration is the arithmetic register size and RAM for the accumulated metric. It can, however, be shown that for our particular metric increments the maximum difference in metrics for the saved paths during a computation cycle is given by (max branch metric) x $(\frac{K-k}{k})$ which for the worst cast (R=1/2, K=7) equals 84. As a consequence one can always perform a normalization by storing the minimum of the metrics for the saved paths and using this value for subtracting from all the new metrics on the next cycle. In this way we have no register overflow in the metric accumulator.

C. Implementation of the Viterbi Decoder

The Viterbi decoder algorithm is readily programmed and accordingly it is exhibited in a flowchart form in Fig. 13. Since the maximum input chip rate is only 300 chips/sec (for the 3/4 rate) it makes sense to consider utilizing a simple computer architecture to execute the program. We have, however, implemented a similar structure which is more properly classified as a microprogrammed controller and is shown in Fig. 14. The program for the Viterbi algorithm has been programmed in the master control read only memory (CROM). This memory has thirty-two 48-bit words although only 29 of these words are utilized. The various registers, multiplexers and counters are enabled by individual lines or small groups of lines from this CROM. The program is executed by sequencing through the 29 CROM words which are addressed by the program counter which consists of an up counter that periodically has

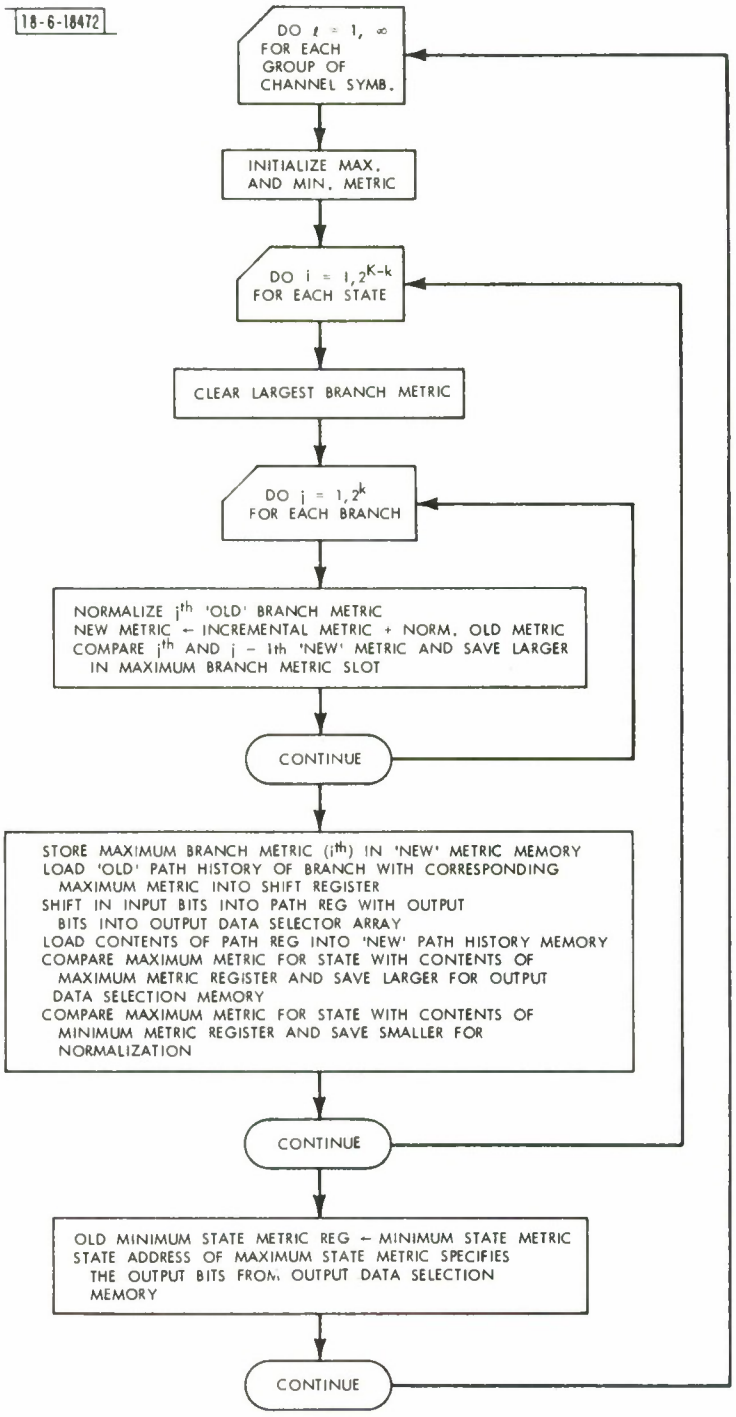


Fig. 13. Flowchart for Viterbi Decoder.

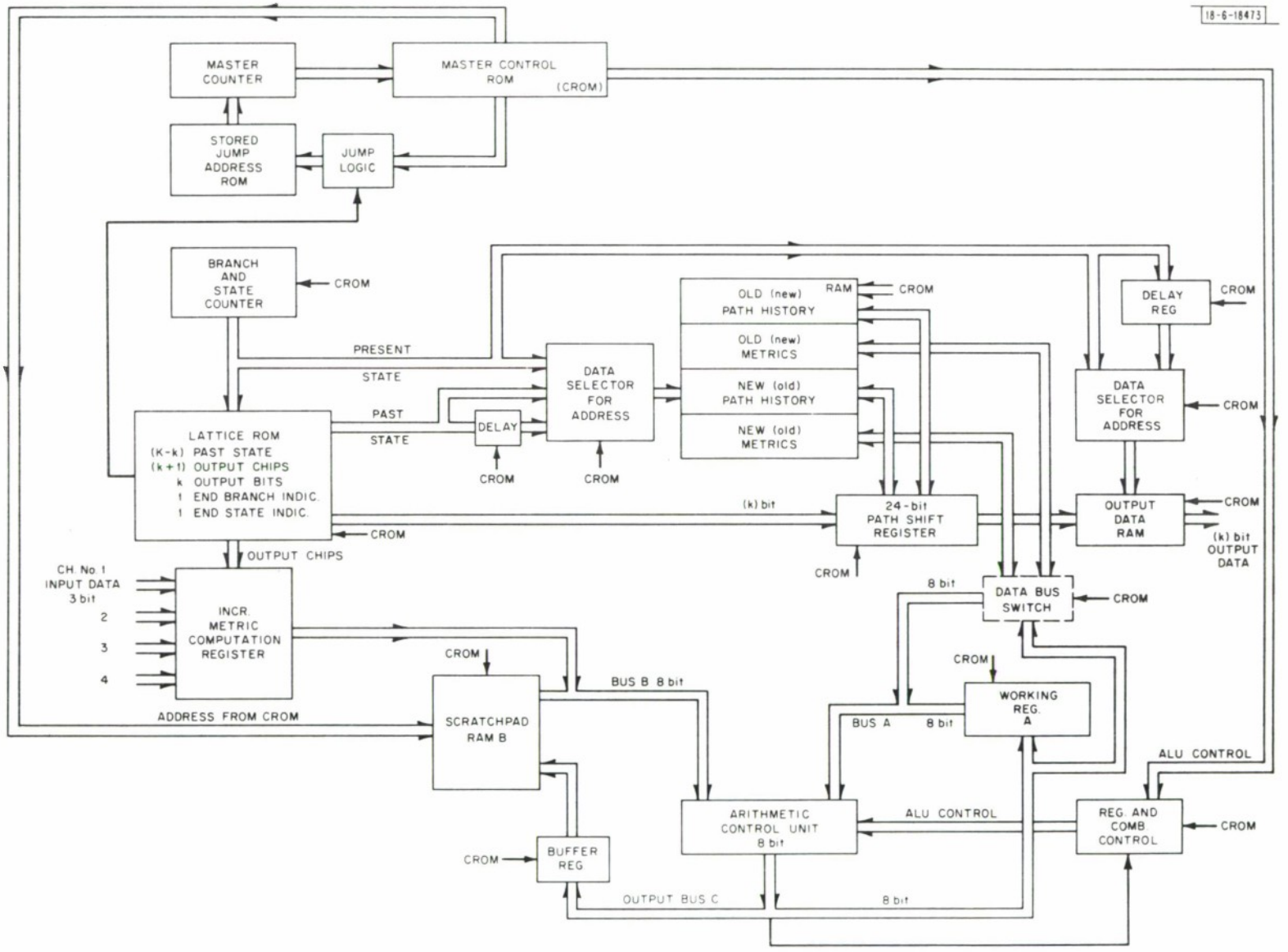


Fig. 14. Viterbi Decoder Implementation.

a jump address loaded, thereby providing a branch or loop capability. The decision to jump depends both on certain bits in the CROM and certain external conditions.

In the Viterbi algorithm the various state transitions (and their associated input bits as well as output channel symbols) for the encoder is examined in a fixed and predetermined sequence. A simple way to accomplish this is to program it in a lattice ROM which is addressed by a counter. The counter address for this is divided into two segments, with the most significant bits representing the state bits and the least significant bits the branch bits. The constraint lengths for the R=1/2, 2/3 and 3/4 codes were selected to be K=7, 8 and 9 respectively so that the number of states (2^{K-k}) turns out to be 64 in all cases. Hence only the number of branches entering each state (2^{k}) is variable and depends on the code rate. The addressing for all rates is easily handled with a single counter and a three to one multiplexer bank. Each word in the lattice ROM gives the prior state, along with the branch's associated input bits and output chips plus flag bits denoting the last branch for each of the states and the last branch for the last state. These flag bits are used in the jump logic for looping. The output chips are compared with the received chips in the incremental metric computation logic resulting in the test incremental metric. The associated input bits are used in constructing the "paths". It is useful to observe that the present state comes directly from the branch and state counter whereas the past state comes from the 'lattice' ROM. Obviously every code rate has its own set of lattice ROMs.

The RAM has been distributed according to function. For example we have a 64 by k bit output data RAM. This k bit word memory segment stores the path bits which were shifted out of the 64 saved and updated path segments. The RAM is written into with the present state as an address and read out once per cycle with that present state with the current maximum metric as an address. This particular state was stored with the conditionally clocked delay register. We also have a 4 word (8 bits per word) scratchpad RAM which stores the 'old' minimum metric (for normalization), the largest branch metric, the 'new' minimum metric and the 'new' maximum metric. The remaining RAM segment contains the old and new state metrics and path history segments. This is a 256 x 24 RAM which is divided up into old and new segments, the location of old and new segments switching every cycle (i.e., every 1/75 sec). Observe that one can address the RAM by either the present state, past state or that particular past state whose metric plus the incremental branch metric equals the maximum for the currently examined present state. (There are three other address bits from the CROM and the cycle toggle flip-flop which select old or new and path or metric segments.)

Unlike a computer we have three working arithmentic logic units. The 8 bit arithmetic control unit for metric manipulation is the most general one with 5 control lines which emanate either from the CROM or from conditional logic depending on the MSB of the control units output and some CROM lines. The other two logic units are highly specialized. The 24 bit path shift register can only load parallel and shift right k bits and hence it can do all the path segments manipulations required. The incremental metric computation

unit consists of the previously noted metric mapping units plus simple binary adders and a final holding register which in turn feeds bus B.

Lastly it should be noted that the clocking of all the registers occurs with clocks that are combinatorial outputs of the master clock (600 KHz) and particular CROM bits. Thus all the register clocks occur simultaneously (when required) and race problems are eliminated. The control program was easy to write in 1's and 0's since there were only 29 words. (It is possible to use slightly fewer words with more careful attention paid to the read-write control signals to the RAMs.) With careful selection of chips it may be possible to realize this circuit with somewhere between 50 to 60 chips. Our realization contains nearer 80 chips because we have put in some test features and haven't attempted to strictly economize on the number of chips.

V. DATA COLLECTION FACILITY

A. Introduction

In order to properly assess the modified receiver's behavior in the presence of scintillation we have assembled a data collection facility in which the data storage medium is a conventional digital magnetic tape unit. The controller for this tape drive is a specially constructed unit which is built around INTEL's 8080 eight-bit microprocessor chip. This controller accepts data from a multiplicity of inputs, formats the data and sends the data to the magnetic tape buffer for recording. The controller organization can be seen in Fig. 15 and its connections in Fig. 10. The organization of the controller is similar to that of a small computer. A PROM bank contains a number of programs which can be accessed via teletype. The basic programs are the Operating System Monitor, a tape WRITE program, a tape FETCH program and a record READ program. The Operating System Monitor allows us to start program execution at a selected location, read PROM and both read and modify registers and RAM. The tape WRITE program collects the incoming data and formats the data into structured 1024 byte records which are transferred to tape. The FETCH program in conjunction with the READ program allows one to locate a desired record from the tape and display any selected portion of the record on the teletype. This feature is used to validate proper operation of the collection facility in addition to performing very sketchy data analysis in the field.

The system was developed with the use of an Intellec Mod 80 Developmental System which has the software support to enable a quick development of this moderately complex controller. In the interest of rapid development we decided to use Intel's CPU, RAM, ROM and I/O cards. This suggested construction of a rack

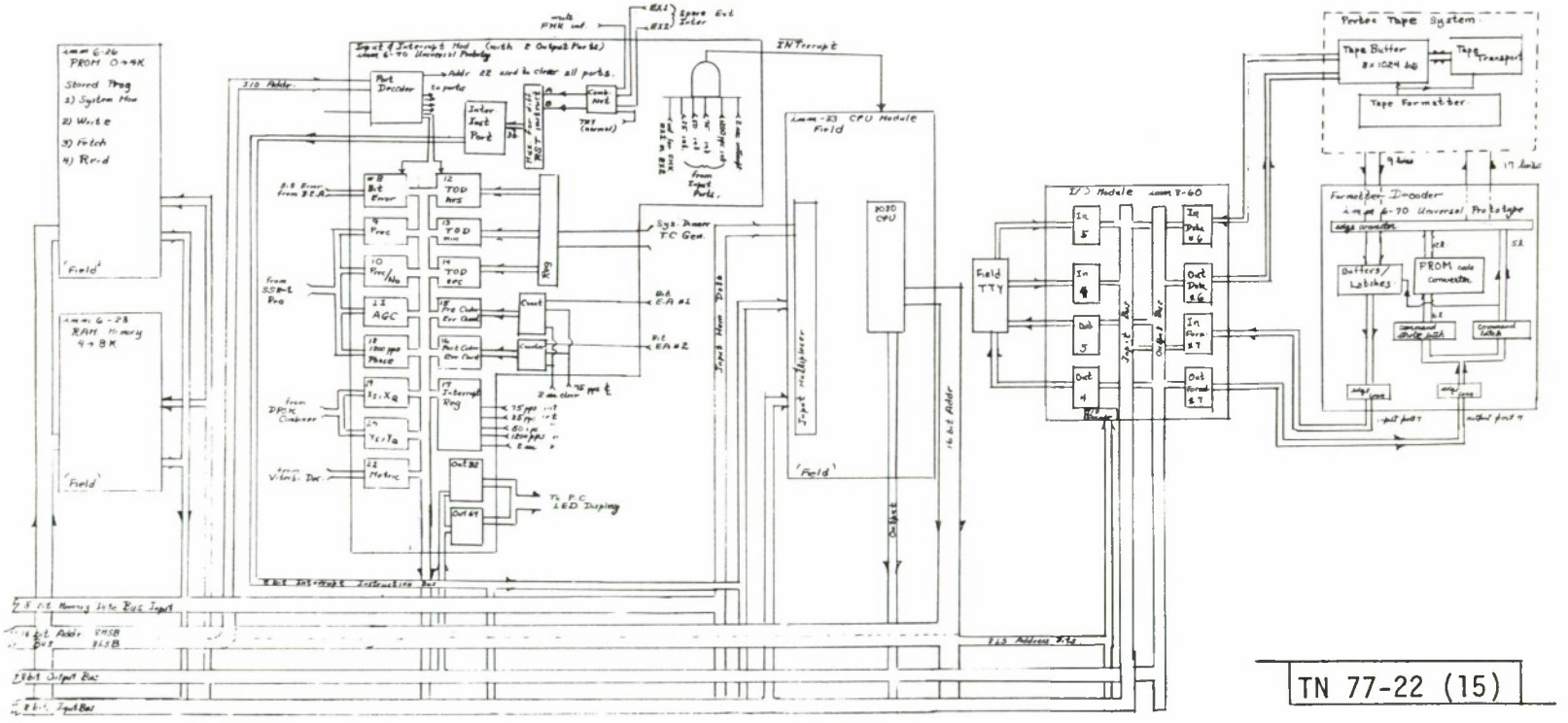


Fig. 15. Data Collection System Organization (Showing Developmental System Connections).

with an (almost) identical bus structure to the Intellec. We did however develop two extra cards, a formatter-decoder and a special input card which also includes the interrupt hardware. Since there is only one input command [In (port number)] and one output command [Out (port number)] it is necessary to convert from the 8-bit bytes which is utilized by the CPU to the multibit and sequential patterns required by the tape buffer. The formatter decoder card performs this function of multiplexing (or demultiplexing) and decoding of the handshaking commands that properly interface the controller to the tape transport. The special input card has 14 input ports (and 2 output ports) and is the main interface between the UHF receiver inputs and the temporary RAM storage. The other I/O card interfaces the CPU to the ASR-33 teletype and the formatter - decoder card.

B. Interrupt System

We have implemented a simple interrupt scheme in conjunction with the 'WRITE' program. There are three non-synchronous interrupts $\overline{\text{WFMK}}$ (write file mark) and as yet two unassigned extras $\overline{\text{EX1}}$ and $\overline{\text{EX2}}$ all of which occur on depressing front panel momentary switches. When $\overline{\text{WFMK}}$ is selected, an $\overline{\text{INT}}$ pulse is generated and sent to the CPU board. The CPU board then inputs the interrupt instruction port which will then contain the instruction RST 3 which in turn transfers control from the main running program to the subroutine LWD which writes a file mark on tape and then places the system offline. (This normally is used to conclude every data tape.)

The main interrupts are synchronous and we use a simple polling technique to select the subroutines we require for writing records. There is a dominant interrupt occurring every 2 sec which defines a record and secondary interrupts occurring at 1/25, 1/50, 1/75 and 1/1200 sec intervals. It should be noted

that these interrupts were generated so as to all occur simultaneously on the 2 sec mark. (See Fig. 16.) This maximizes the minimum time separation between any two interrupts to 1/1200 sec = 833 μsec, thus allowing us to service without interruption a number of data collection routines. The 1200 pps interrupts are for collecting phase information and only the first 150 of these interrupts are utilized in every 2 sec interval (this requires 1/8 sec). A tape record is comprised of a 24-byte header and fifty subrecords of 20 bytes each. A subrecord is written during each 1/25 sec interval (note 1/25 x 50 = 2 sec). The interrupt pulses strobe the input ports which in turn generate interrupt levels which are strobed onto the interrupt register port (17) with a 1/1200 sec clock. The contents of this register is polled bit by bit and subroutines selected on the basis of the value of the bits. For example the 2 sec interrupt starts the program which first stores in RAM the necessary header information and then outputs the information onto the magnetic tape. The 1/75 sec interrupt starts a subroutine that writes into RAM the current values of X_I, Y_I, X_Q, Y_Q, max metric and bit errors. Similar subroutines are available for use with the 1/25, 1/50 and 1/1200 sec interrupts. It should be noted that after the 2 sec interrupt pulse the header is written within the next 2/75 sec. After the 3rd 1/75 sec interrupt is serviced, a subrecord write routine, All, is called which writes the subrecord from RAM to the tape. It should be noted that subrecords are associated with the 1/25 sec interrupts. The subrecords are always written after servicing the third 1/75 interrupt following the 1/25 sec interrupt.

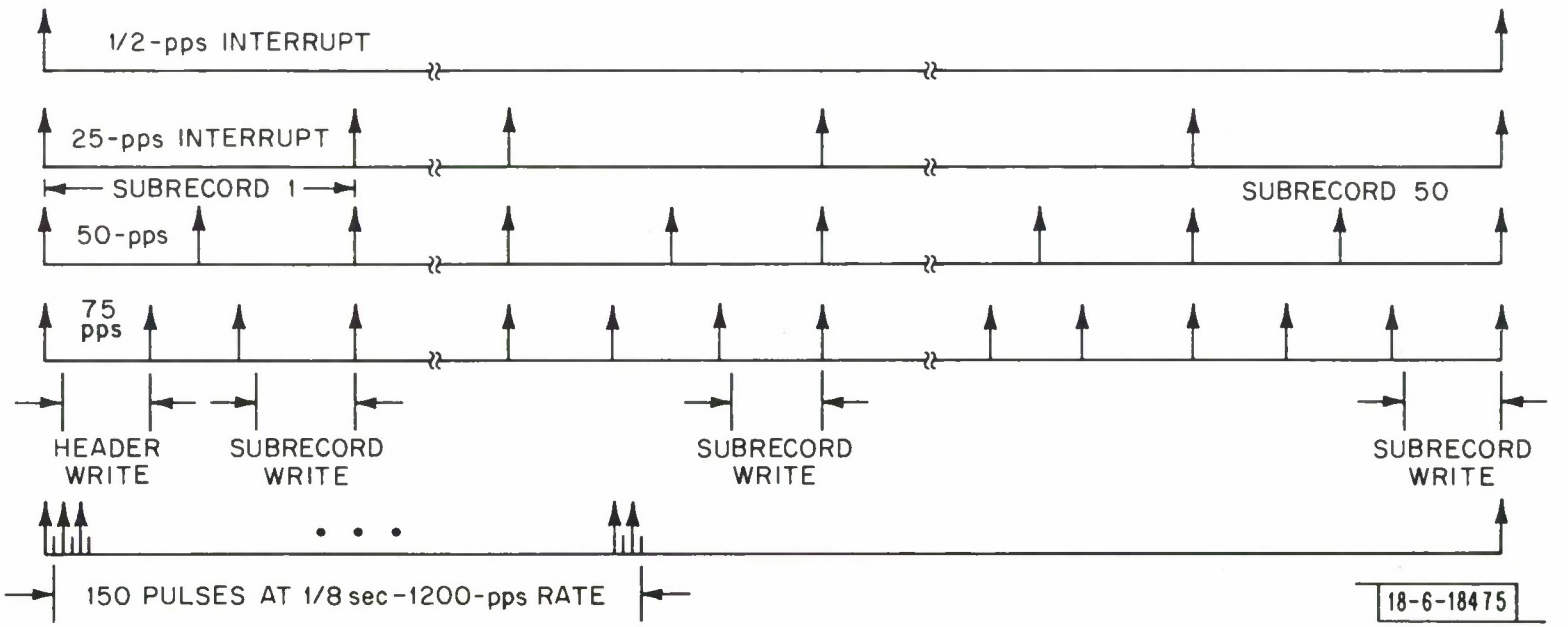


Fig. 16. Interrupt Pulses and Record Structure.

The input and output ports are addressable by the eight most significant bits of the address (MAD8 → MAD15). All the ports are implemented with a very versatile MSI chip (Intel 8212) which is configured in a number of ways. Ports 64 and 128 are configured as output latches enabled by the signal $\overline{\text{I/O OUT}}$. Input ports 12, 13, 14, 15 and 16 are configured as input latches enabled by the encoder. Input ports 8 - 11 and 17 - 21 are set up as interrupting input ports. Observe that these ports are cleared with the IN 22 command and strobed by the various interrupt pulses. On receipt of an interrupt pulse from the receiver the input port will respond with an interrupt level '0' which disappears if either a clear command is given (through IN 22) or after that data on that port has been transmitted to the I/O bus (i.e., the port has been selected). In the WRITE program we use the IN 22 only to initialize conditions at the tape beginning and thereafter use port selection to clear the interrupt levels. It should be apparent that we want to remove these interrupt levels so that we will be ready to accept the next interrupt pulses without having residual interrupt levels existing from a previous set of interrupts. It is noted that the interrupt levels are ANDed together to give the output $\overline{\text{INT}}$ which goes to the CPU board (via pin 42).

The 1812 is used as an interrupting input port in the following manner (refer to Fig. 17). Tie MD = 0 and hence a positive going interrupt pulse on the STB first clocks in the data on the D flip-flop latches. (Assume initially $\overline{\text{CLR}}$ = 1 and $\overline{\text{INT}}$ = 1. The strobe also will clock the SR flip-flop putting Q_{SR} = 0 which drives $\overline{\text{INT}}$ → 0. Now when the port is selected $\overline{\text{DS1}} \cdot \text{DS2} \rightarrow 1$, the eight-line output is dumped on the output bus, $\overline{\text{INT}}$ is forced to remain at 0

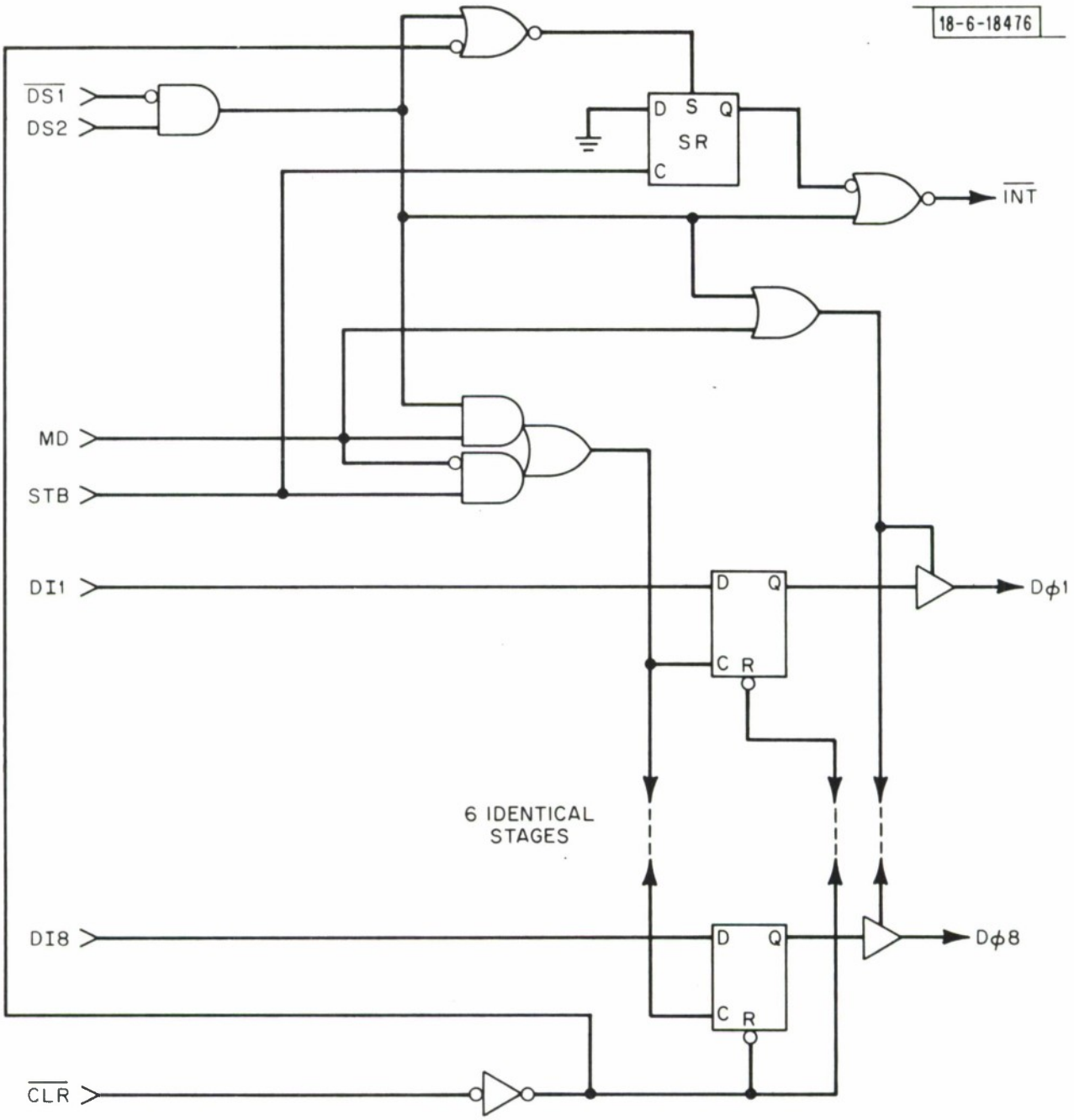


Fig. 17. I/O Port (Intel 8212).

and, most importantly, will set the SR flip flop so that $Q_{SR} \rightarrow 1$. Then when $\overline{DS1} \cdot DS2 \rightarrow 0$ the interrupt $\overline{INT} \rightarrow 1$ which indicates that the interrupt is now serviced and cleared. The only other way to clear the interrupt is to set $\overline{CRL} \rightarrow 0$. Observe that the interrupt out of the interrupt ports (which feed the composite interrupt gate A9 as well as the input to the interrupt register port 17) are triggered by interrupt strobes from the receiver timing. This also means that in order to clock in the interrupt levels appropriately one must use a strobe for port 17 that is delayed until the ports have time to respond. Alternatively one can use the port as a straight-through buffer with no strobe since the interrupt lines should be self clearing.

C. Important Programming Details of the Interrupt Service Routines

The WRITE program is accessed with a G0310 command from the system monitor. See Fig. 18. At this time the external interrupt pulses are not recognized by the CPU since the DI command was given in the first step of the system monitor program. The WRITE program provides a 4-inch space after the B. O. T. mark, writes the beginning filemark, puts the formatter into the autowrite continuous mode, prints 'READY,' clears the interrupt levels (by clearing input ports) and goes into a loop which searches for the 2 sec interrupt level. At this point the interrupt pulses should be switched on externally. Then the service routine TRY is called: the address of the label HALT1 is stored in the stack and when TRY's final routine (A9) is completely serviced the RET instruction will put the system into the HALT1 waiting loop (a two-instruction loop). It should be mentioned that TRY consists of a number of uninterrupted subroutines (PBEG, P1200, P25, P50, and P75)

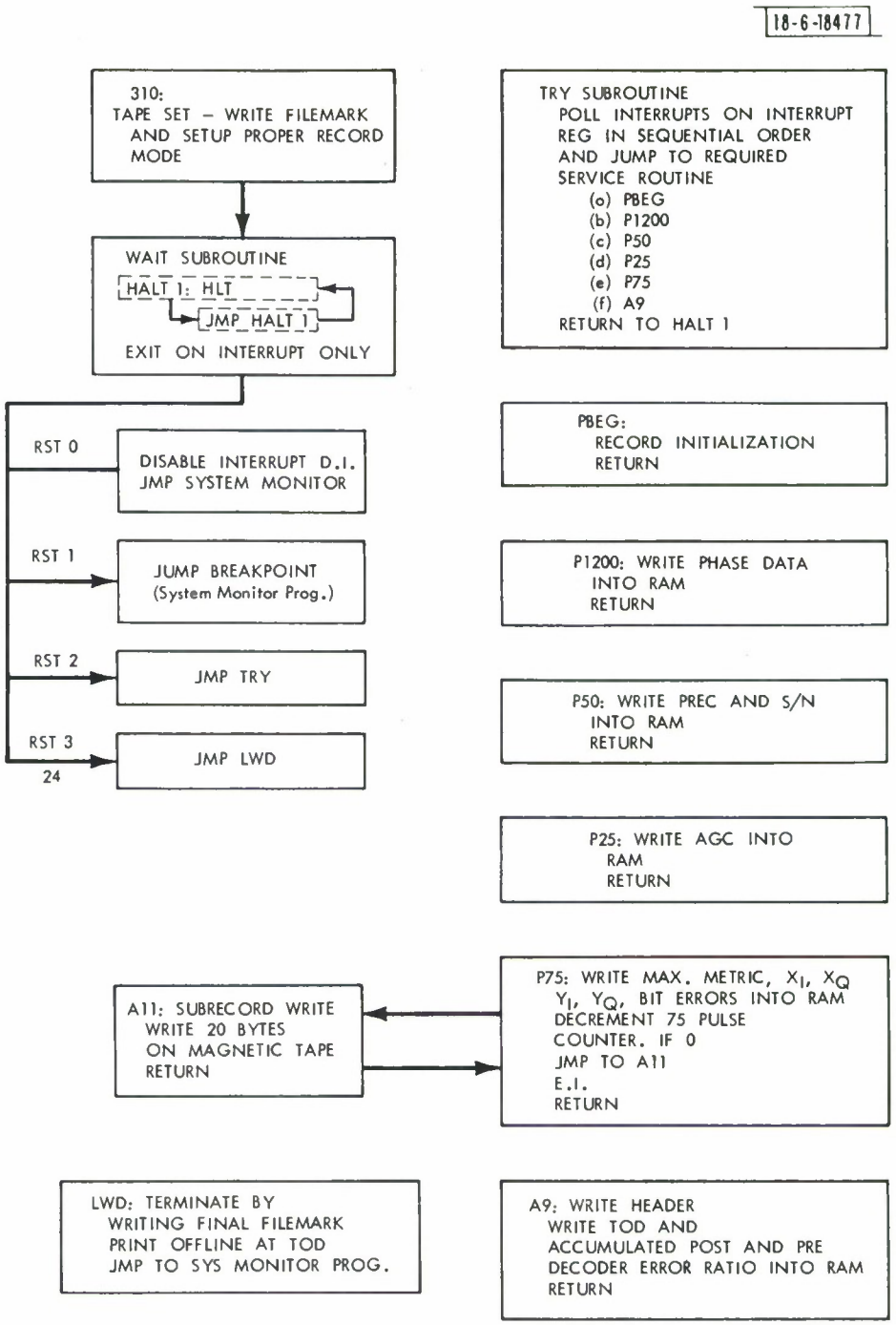


Fig. 18. WRITE Flowchart (Simplified).

all of which must be serviced before the next interrupt occurs. However, the subroutines A9 and A11 (which are also a part of TRY) are too lengthy and must be interrupted at least once in order to service PBDG through part of P75. It should be emphasized that A9 must be completely serviced once every 2 sec although it can and will be interrupted during its execution. For this reason we require an enable interrupt every time we go into TRY.

The following sequence of operation occurs following the first 2 sec interrupt pulse (which the first time does not actually interrupt the CPU since DI is set on the first step of the program.)

1) Program TRY is called and we go through and execute PBDG through P75.

2) Subroutine A9 is automatically entered and execution begins and is terminated by an interrupt. The PC at this point is put into the stack as a result of the interrupt instruction RST2. (Note that the RST instruction acts as an implicit call which require a RET to get back to the original interrupted program sequence.) We then start re-executing the TRY program where we now execute PBDG through P75, skip A9, finish the rest of the TRY routine and then RET pops from the stack to the PC the location at which we interrupted the A9 routine originally.

3) A9 is then in process of execution and if we have another interrupt before A9 completes execution, the same procedure as outlined in 2) follows. If, however, execution of A9 is completed we then want to enter the HALT1 waiting loop. This is automatically done the first time since the RET goes to the instruction following the CALL TRY instruction at location 0370H (which is HALT1: HLT).

4) The HALT1: loop is exited only with the use of interrupts which put us into another CALL TRY command via the RST2 instruction which terminates the TRY routine then puts us automatically into the HALT1 waiting loop at the JMP HALT1 instruction.

On the second 2 sec interrupt pulse we come out of the HALT1 loop and will, on the RST 2 command, again put the HALT1 label on the stack. The RET statement at the end of TRY routine will put us back into the HALT1 loop.

Notice that there is an EI command following the P75 routine. This enables us to accept an interrupt during execution of A9 or A11 so that the necessary time dependent routines PBEG → P75 can be serviced. When A9 or A11 is interrupted the effect of the RST 2 instruction is to load the stack with the address in A9 or A11 which we want to return to after we perform the service routines. The first instructions in the TRY routine also PUSH the registers on the stack so that prior to reentering A9 or A11 we restore the registers with the appropriate POP instructions. When the service routine is being executed after interrupt of A9 (or A11) we do not want to start executing A9 (or A11) at the beginning again but want to jump down to the last few statements in TRY, POP the register appropriate to that stage of A9 (or A11) execution and the RET should then start the program execution at the stop following the point in A9 (or A11) where execution was interrupted.

A9 is called only once every 2 sec by the following artifice. During PBEG we set an originally grounded interrupt register bit equal to 1. The conditional call CC A9 responds to this bit only once per 2 sec.

All is also called once every 1/25 sec after collecting the data from the 3rd 1/75 interrupt pulse. (The first 1/75 sec interrupt occurs simutaneously with the 2 sec interrupt and remains in sync from then on.) The subrecord write program All then is executed in the 1/75 sec interval prior to the next 1/25 interrupt pulse (which also coincides with the next (first) 1/75 sec interrupt on the next subrecord). The subrecord write program, All, is of course allowed to be interrupted and needs only to finish execution prior to the 1/25 sec pulse.

D. Field System Monitor Program

At present the field system monitor program is most useful when used in the developmental system which includes the MOD 80 unit. The developmental system contains a RAM memory and we thus are allowed to modify instructions. One of the monitor commands is G address 1, address 2 where "address 1" is the address where we want to start execution and "address 2" is the address at which we want to set a breakpoint. If we don't set a breakpoint then we just start execution at "address 1". If a breakpoint is selected the contents of that breakpoint address is stored and replaced with a RST 1 instruction. Next we start program execution at "address 1" and when the breakpoint is reached the RST 1 instruction puts us into the BREAl routine which stores the current register contents and puts us back at the beginning of the system monitor so memory can be interrogated and values substituted if one wishes with the S command or interrogate and change register values with the 'X' command. (Prior to getting out of the BREAl routine we restore the original contents of the breakpoint address.) After suitable interrogation and change

we then can continue if one wishes with another G address 1, address 2 where now "address 1" was the original B.P. and "address 2" is an optional new B.P.

It would be beneficial to have the same flexibility when the programs have been converted to EPROM. However, note that one is not able to inject a RST 1 instruction at the breakpoint address (since obviously one cannot write into EPROM). If the PC were accessible, it would be possible that one could compare this value with the breakpoint address and branch accordingly when equal values occur. However, this would require comparisons after every instruction and certainly would increase the execution time such that real time operation with our system of interrupts would not be possible. (As a point of fact the PC is not recoverable with the instruction set available.) Likewise, simple hardware solutions are not apparent.

VI. SYSTEM EVALUATION

A. Laboratory Simulated Link Test

Upon completion of the construction of the individual subsystems it became necessary to test the composite over a channel that is a good approximation to the actual satellite downlink (the uplink being assumed uncorrupted). Accordingly the test setup was constructed and is shown in Fig. 19. This required the construction of a number of additional "test" units. A test multiplexer was built to perform the same function as the Navy's TDM 1150 (time division multiplexer). The purpose of the multiplexer is to combine sixteen 75 bps channels (15 data streams and a frame synchronization stream) into a 1200 bps stream. After DPSK encoding, the stream biphase modulates a UHF signal. The simple modulator unit is the counterpart of the Navy's WSC-5 transmitter. The channel fading is approximated via a voltage controlled attenuator which is driven by a Lincoln Laboratory built Rayleigh simulator. An additional step attenuator is inserted in the line so that one can vary the $\overline{E_b}/N_o$ ratio over a considerable range. The output of the final attenuator is fed into the amplifier-converter unit of the scintillation receiver and recording system. Bit errors both prior to and following the decoder are detected by the error analyzers. It is noted that the pseudo-random pattern generator and the bit error analyzer are merely the transmitter and receiver sections of a HP 1645A error analyzer.

With the above setup it is easy to perform probability of error versus $\overline{E_b}/N_o$ measurements for the selected coding alternative. Any one of the combinations from Table III can be then evaluated.

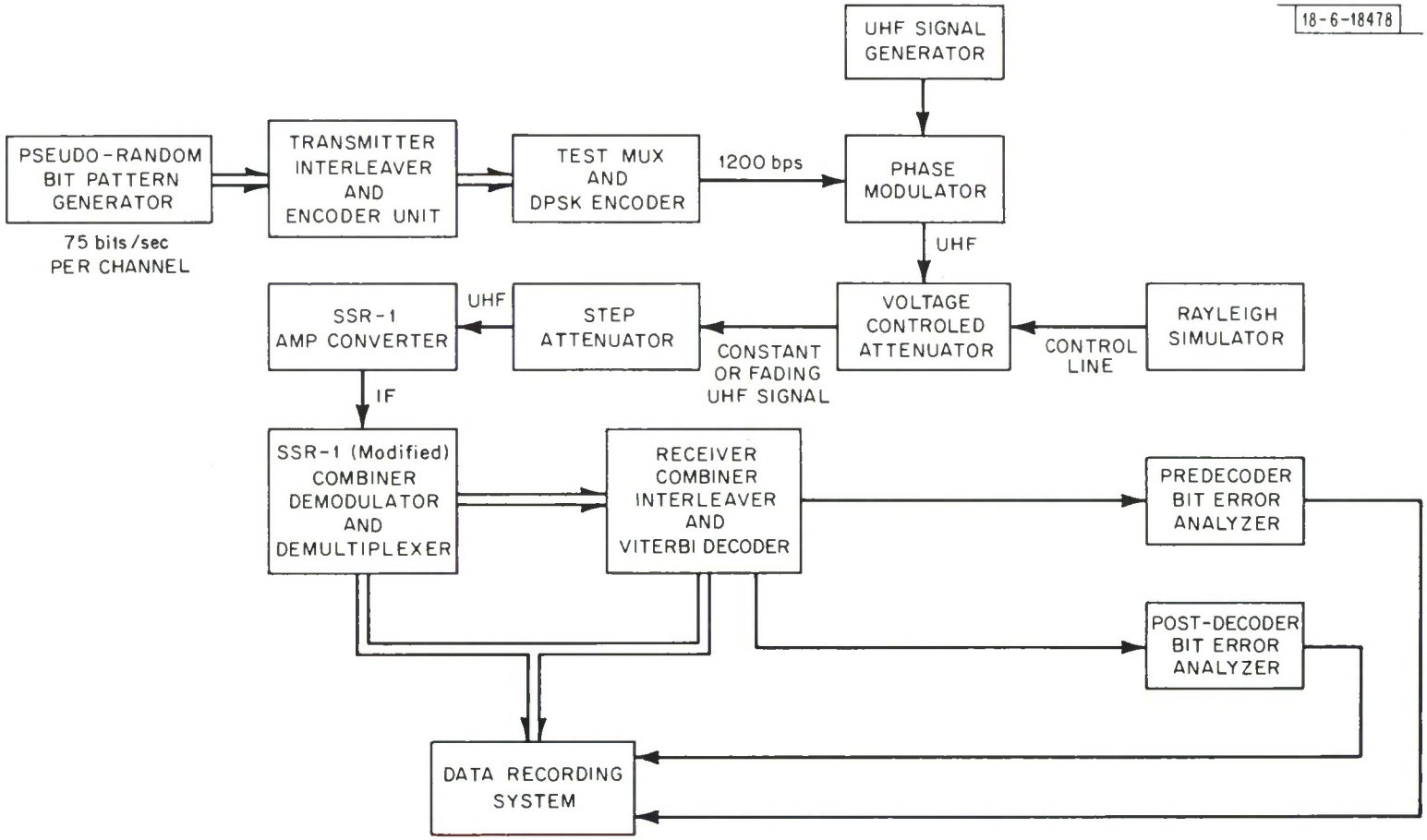


Fig. 19. Laboratory Link Simulation Experiment.

TABLE III
RECEIVER CHOICES

a) Systematic or nonsystematic encoding

b) Code rate 1/2, 2/3 or 3/4

c) Coherent or differentially coherent combining

d) Gaussian noise plus constant signal or Gaussian noise plus Rayleigh fading signals

A complete analysis would result in 24 sets of curves. The 12 sets of curves for Gaussian noise alone are less useful and are mainly used to check the system against known results from a multitude of investigators.

It would first appear that deriving the (P_E, E_b/N_o) experimental curves for the case where the input signal is only corrupted by Gaussian noise (no fading) would be quite easy. One could then compare these with the theoretical curves published in the literature. Unfortunately there are a number of factors which complicate the issue. First and foremost is the fact that the single channel information bit rate is only 75 bps (or about 2.7×10^5 bit/hour). To measure the error rate of, say, 10^{-5} with any degree of statistical confidence one would need a four hour trial (10^{-6} requires a 40 hour trial). As shown Fig. 20, the theoretical post-decoder error rate curves drop abruptly with increasing E_b/N_o (essentially no reliable data to $P_e \sim 10^{-5}$ with only a 2 dB increase in E_b/N_o). We estimate that our measurement accuracy for E_b/N_o to have an uncertainty of $\pm$ 1/2 dB. From the figure one sees an apparent 1 to 2 dB difference (between theoretical and measured) in the post decoder curves. It should be pointed out that most of this discrepancy is due to the tracking behavior of the SSR-1 receiver. In particular when the channel symbol to

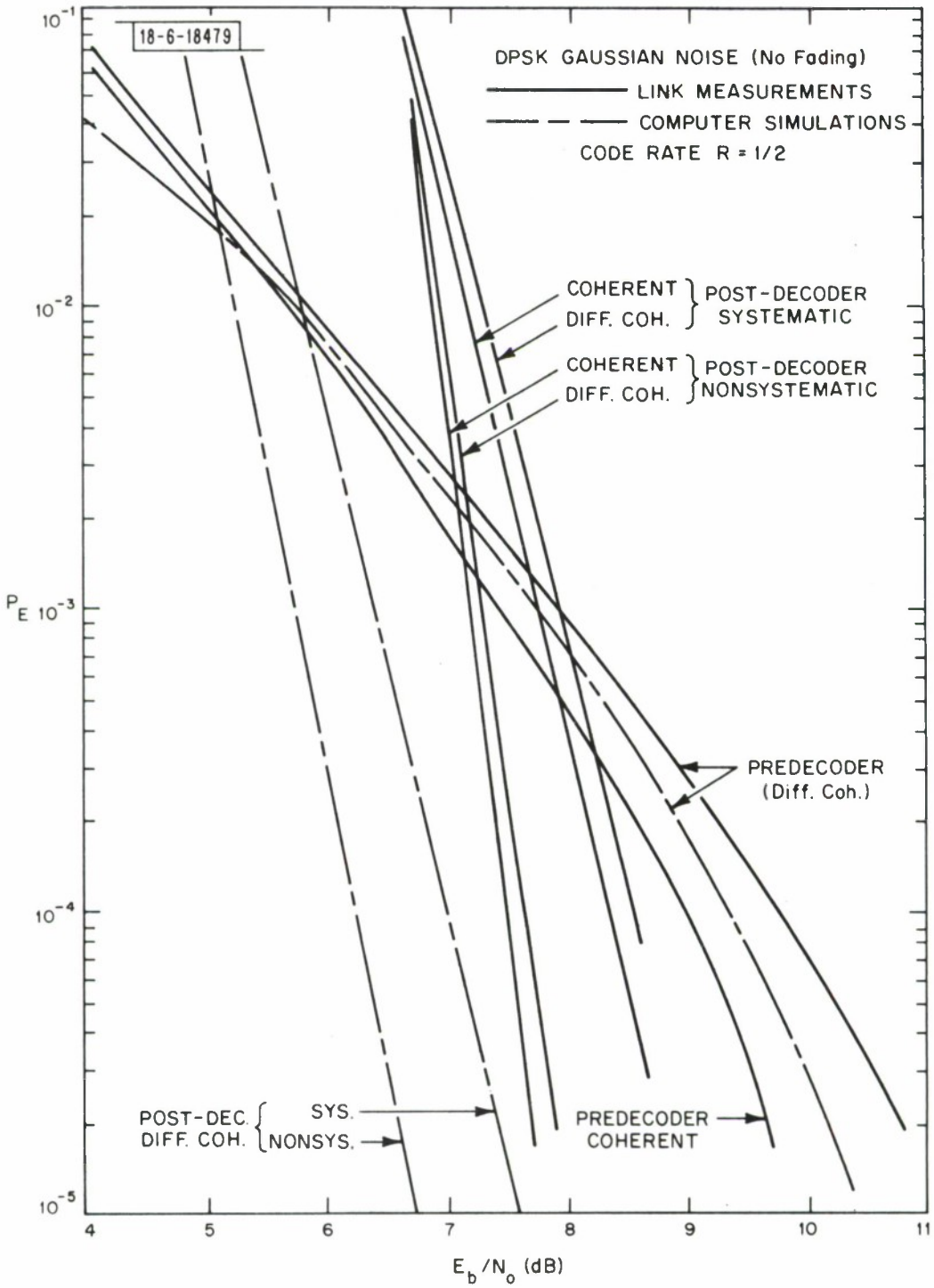


Fig. 20. Error Rate for R = 1/2, DPSK Encoded and no Fading, K = 7.

noise ratio approaches about 4 dB the tracking in the SSR-1 causes significant degradation (for the R = 1/2 code this corresponds to E_b/N_o = 7 dB where E_b is the energy in an information bit as opposed to the energy of the channel symbol E_c). One then can understand why the two sets of curves are separated at the higher P_E values. Because of the steep slope of these curves it was difficult and time consuming to get statistically significant estimates of P_E for values of E_b/N_o between 7 and 8.5 dB. Given the above difficulties, the measurements indicate that at a $P_E \sim 10^{-5}$ the difference is about 1 dB and decreasing for smaller values of P_E. The raw channel error rate curves (pre-decoded) with their more gradual slopes have a closer agreement as would be expected.

Bucher has constructed a rather complete computer simulation of the P_E degradation due to Rayleigh fading as compared to those cases with Gaussian noise alone. The Rayleigh fading probability characteristic approximated that of previously collected scintillation data. The simulation and hardware implementation used the same type of interleavers and same interleaver configuration and of course the same convolutional encoder-decoder combination. The simulation, however, did not accurately reflect some of the behavior patterns exhibited by the SSR-1 during deep and long duration signal fades. Adding Rayleigh fading along with the Gaussian noise gives the results shown in Fig. 21. In addition to the measurement limitations just described, one must account for the even larger discrepancy between the computer simulation results and the link measurements at values of $P_E < 10^{-3}$ (the difference is about 2.2 dB for the slow fade hardware Rayleigh simulator). It must be pointed out that the

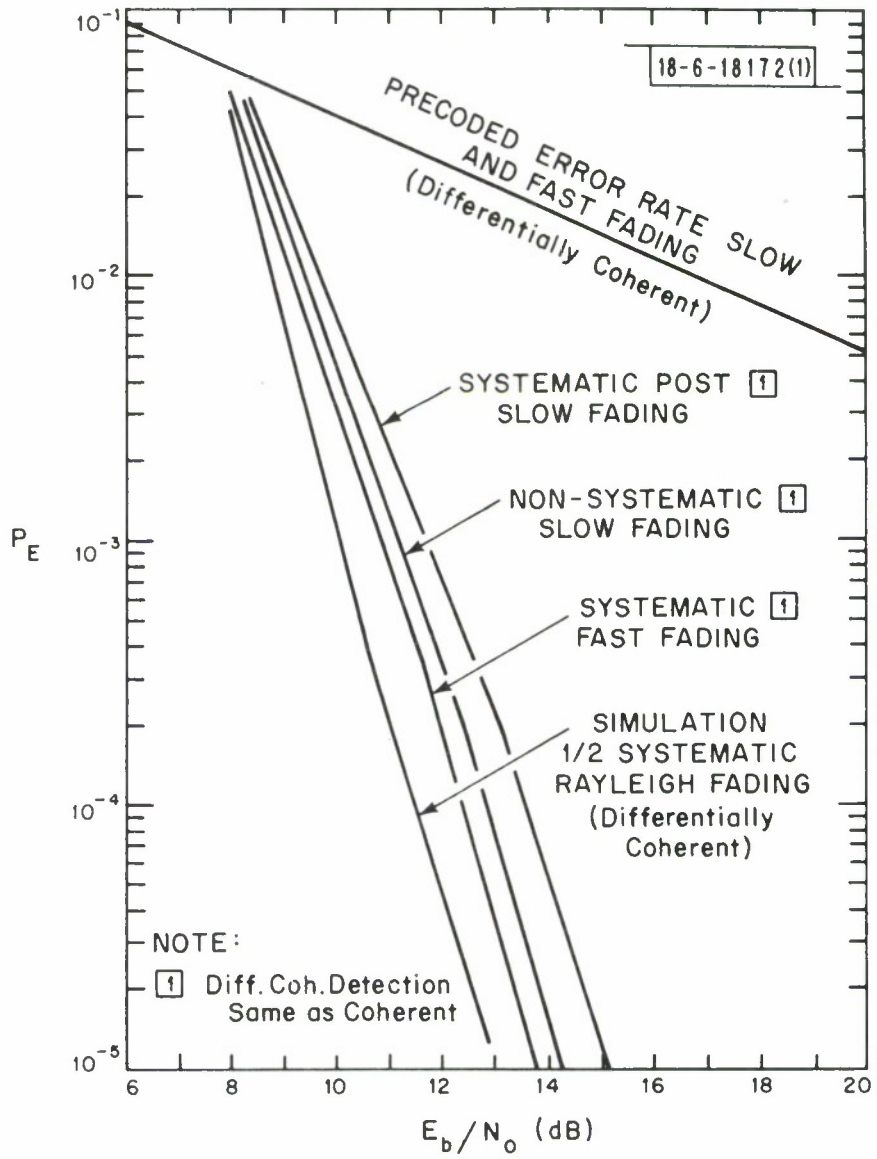


Fig. 21. Error Rate for R = 1/2, DPSK Encoded and Rayleigh Fading, K = 7.

difference results as a consequence of the fact that the Rayleigh fade distribution for the computer simulation (although a close match) was not the same as the Rayleigh fade duration distribution that was generated with hardware. In particular the hardware realization resulted in a tail of the distribution that had a higher probability of long duration fades due apparently to a hardware saturation effect. This was first recognized when the SSR-1 receiver was observed to lose track and slip a bit resulting in a loss of interleaver sync. (This effect prompted us to design a more complex master clock PLL as noted in Appendix 3.) In order to cut down on the probability of long fades we increased the filter bandwidth in the hardware simulator. This resulted in new fade distribution density that had a better approximation for the tail of distribution as compared to the computer simulation model although the main part of the distribution showed a modest skew. The amplitude distribution changed only slightly. The resulting error rate curve on Fig. 21 is identified by the term "systematic post-decoder fast fading." The curve differs by at most 1 dB from the simulation curve. In defense of this attempt at reconciliation it must be recognized that the beneficial effects of interleaving depend to first order on the tails of the fade duration distributions. Moreover to attempt to provide a better match between hardware and computer simulation results would have required a much longer link test period that we could afford. These simulations are very important in that they form a benchmark with which to compare the actual satellite link tests. These comparisons will be reserved for the next report.

B. Field Tests

In order to verify satisfactory operation of all the system interfaces, two sets of tests were conducted with the transmitter encoder and interleaver system located at the Norfolk, Virginia, Navy Communication Area Master Station (CAMS) and the receiver located at Lincoln Laboratory. The October 1975 tests utilized the LES-6 satellite and the March 1976 tests used the Atlantic Marisat. In both cases there was no discernible fading (as expected) and hence the tests were solely for the purpose of identification of interface and operational problems. It should be noted that aside from the problem of nonalignment of data channels (due to the transmitter's TDM 1150 behavior) the complete system worked satisfactorily.

In June of 1976 the transmitter encoder-interleaver was installed at CAMS EASTPAC Honolulu and the receiver at CAMS WESTPAC Guam. These tests use the Pacific Marisat satellite. We were operational from August 30 through January 1, 1977. Daily operations were conducted throughout the fall and early winter during the hours 2300 to 0500 GMT which corresponds to 1900-0100 Guam standard time. This is the period in which maximum scintillation had previously been observed. It was expected that scintillation would occur only on the downlink as Hawaii is not in the equatorial scintillation zone. We also moved the transmitter encoder to Guam and ran tests where both the uplink and downlink are subject to scintillation. Strong or weak scintillation was observed on 30 out of 123 days of observation. The tests proved successful in that typically during scintillation periods a raw channel symbol error rate of 3×10^{-3} (for $\overline{E_b}/N_o \sim 21$ dB) would after decoding result in an information

bit error rate of essentially zero. The results will be detailed in a follow-on report.

The satellite used in the comprehensive field tests was the Pacific Marisat Satellite (Gapfiller). The test broadcast was sent over one of the 25 kHz (narrowband A) channels. The average energy per information bit to noise ratio can be estimated from the following link calculation.

Satellite EIRP	23.0 dBW
Path loss (at 250 MHz)	-172.5 dB
Antenna Gain (1 unobstructed antenna)	0 dB
Power Received	P_r = -149.5 dBW

System noise intensity

receiver noise temperature = 290° K (3 dB N.F.)

antenna temperature $\simeq 200^{\circ}$ K (over hemisphere)

$\sim 490^{\circ}$ K

Equivalent noise intensity N_o = -201.7 dB

P_r/N_o = 52.2 dB

The energy per chip to noise ratio is then (for R = 1200 bps)

$$E_c/N_o = \frac{P_r}{N_o R} = 21.4 \text{ dB}$$

As an example of the improvement provided by our coding scheme consider the case of a R = 1/2 systematic code with coherent processing. The energy per information bit to noise ratio could then be 24.4 dB. From Fig. 21 it can be estimated that the system margin for a post error rate of 10^{-5} is about 10.5 dB (using the experimental fast fading curve as being representative of our system

response to the actual scintillation). At a energy per chip to noise ratio of 21.4 dB, the predecoding error rate is approximately $4x10^{-3}$, which is generally considered unacceptable. Without coding, a SNR of about 40 dB would be required for 10^{-4} bit error rate. Thus in the presence of fading the coded system requires significantly less SNR to provide a reliable link. The final report will cover in detail comparisons of the measured error rates with the curves developed in the link simulations.

The last three figures are photographs of the equipment. Figure 22 shows the overall system including the entire data collecting facility. Figure 23 shows the encoder segment to be used in conjunction with the transmitter. The large number of chips evident is due strictly to having a duplicate system. A single well designed encoder would require no more than 60 I.C.'s for all the rates. Figure 24 shows the decoder-deinterleaver system. Elimination of unnecessary test circuits and use of 4K RAMs instead of 1 K RAMs would eliminate about 50% of the IC's. The entire receiver add-on could be easily put on 2 wire wrap boards and along with the necessary A/D converters mounted in a 3.5 inch drawer. Figure 25 exhibits the analog drawer most of which is associated with the data gathering function. Finally Fig. 26 shows the Intel based controller used in the data collection system.

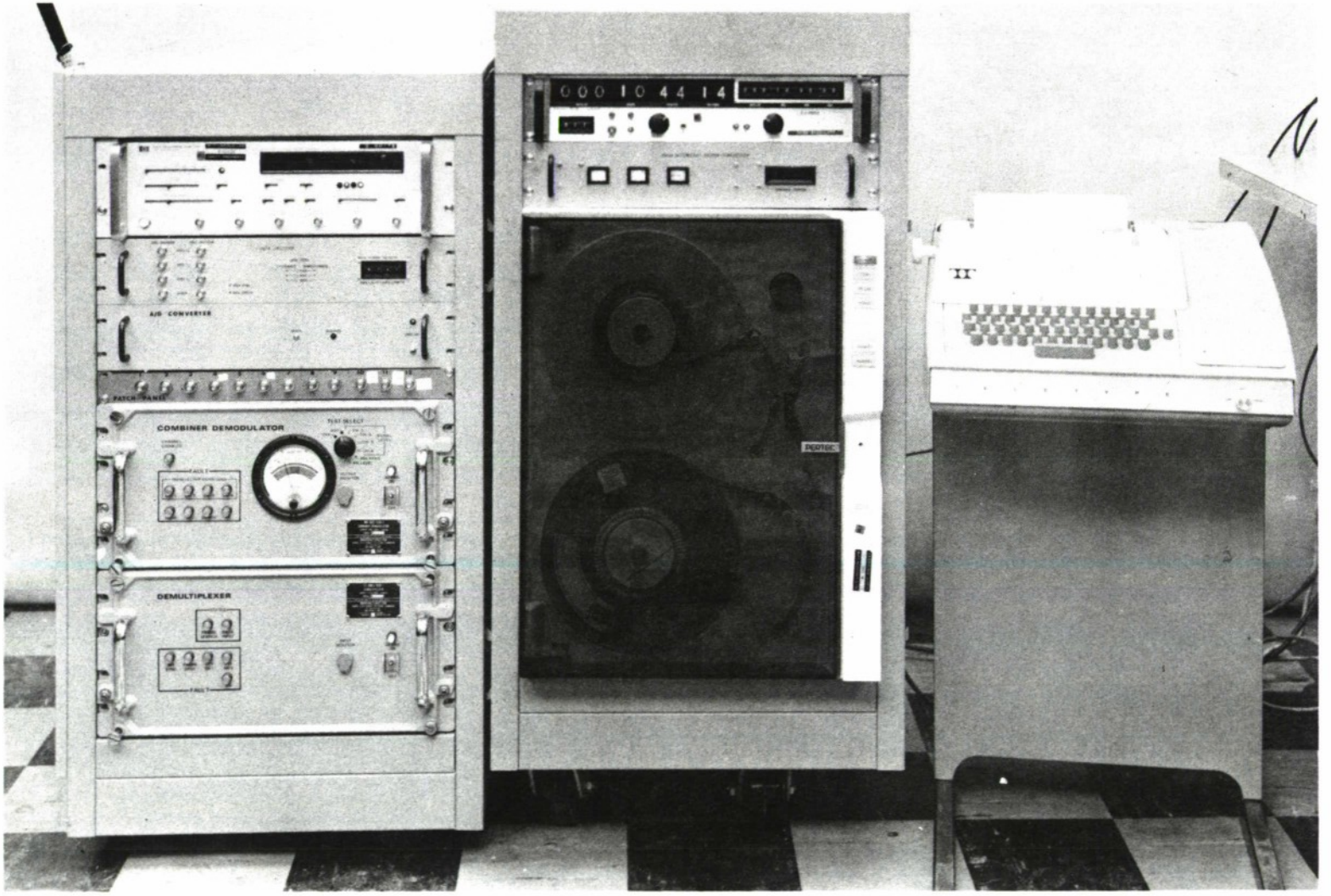


Fig. 22. Scintillation Receiver and Recoding System.

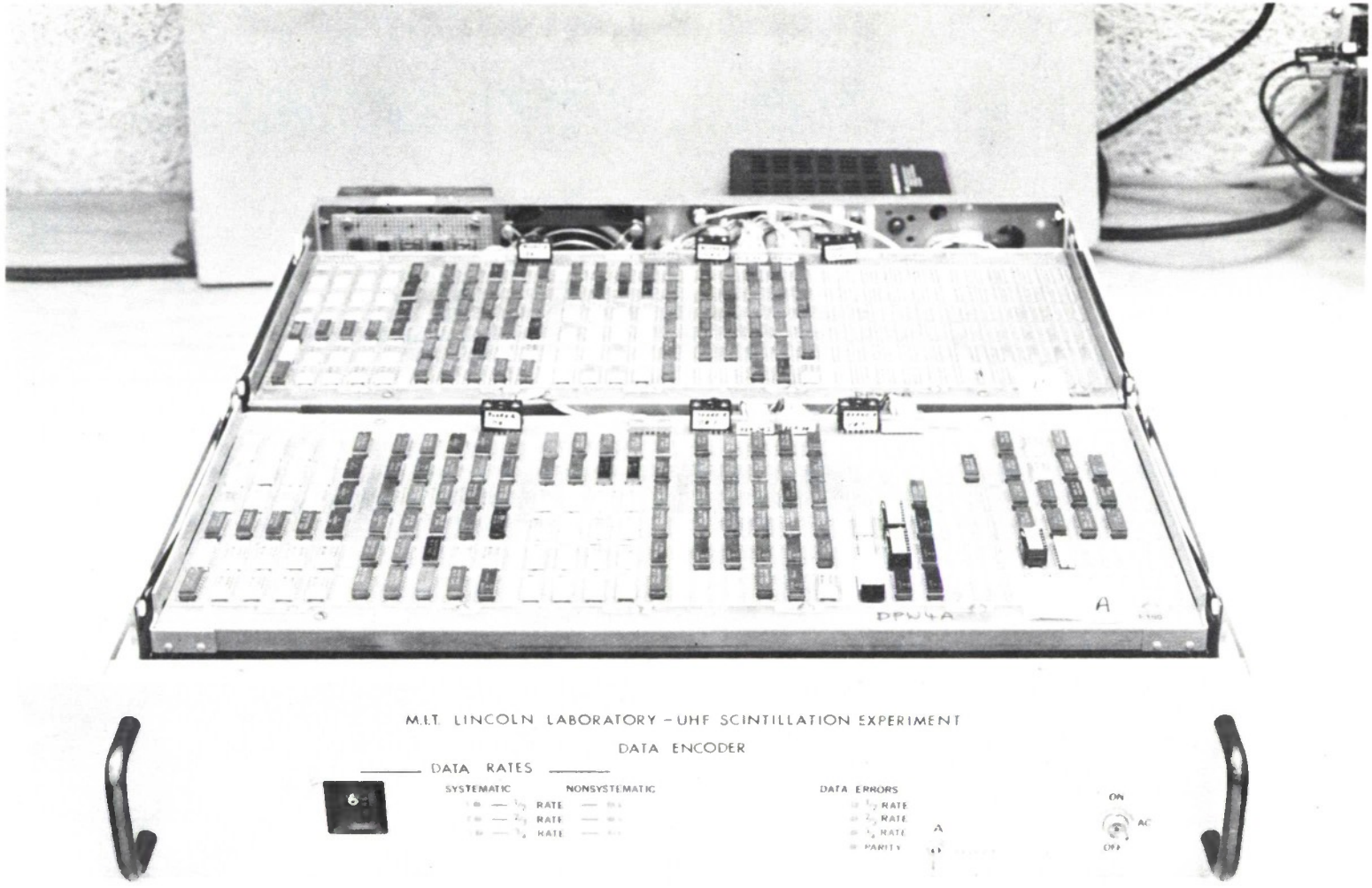


Fig. 23. Transmitter Interleaver-Encoder.

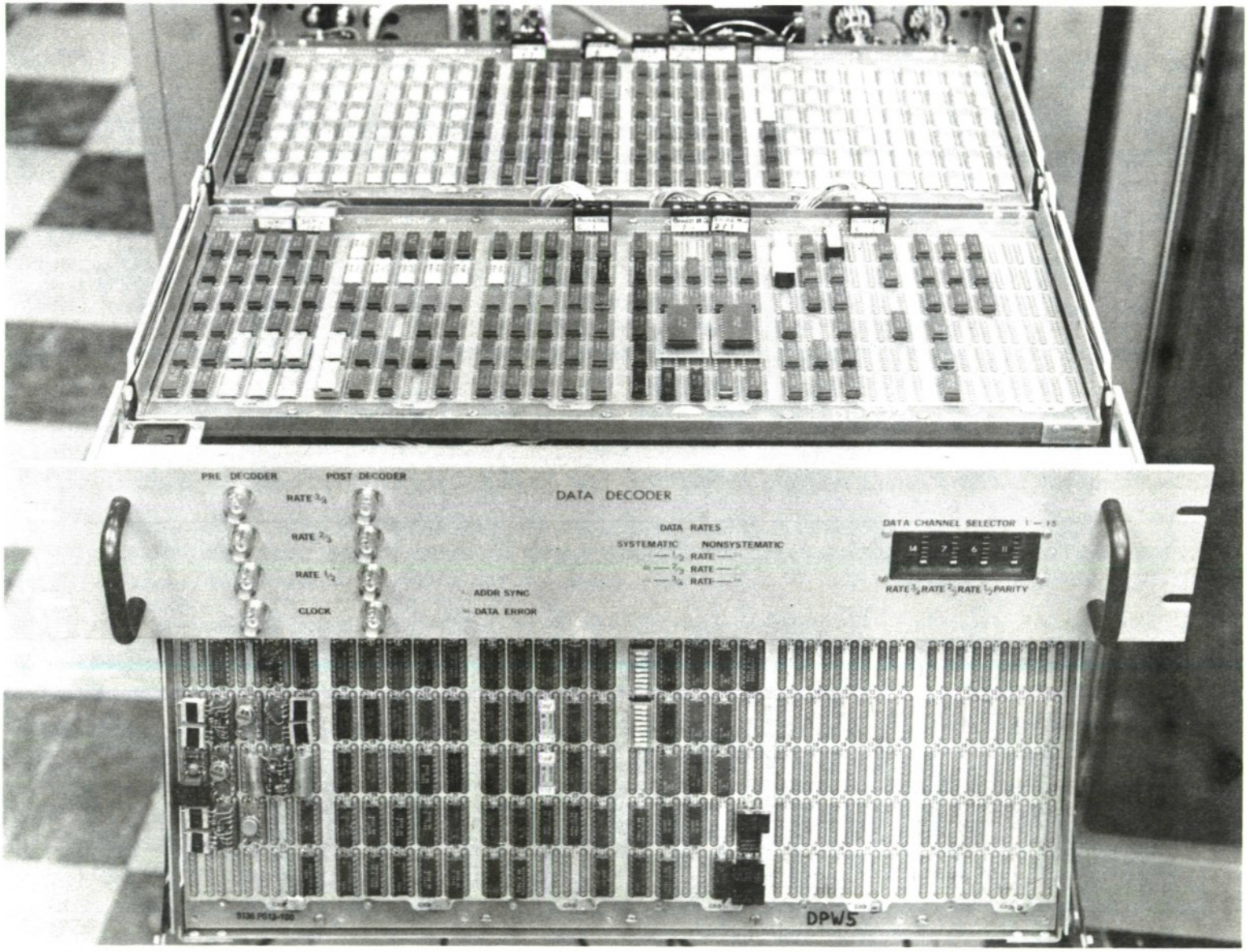


Fig. 24. Receiver Decoder-Deinterleaver.

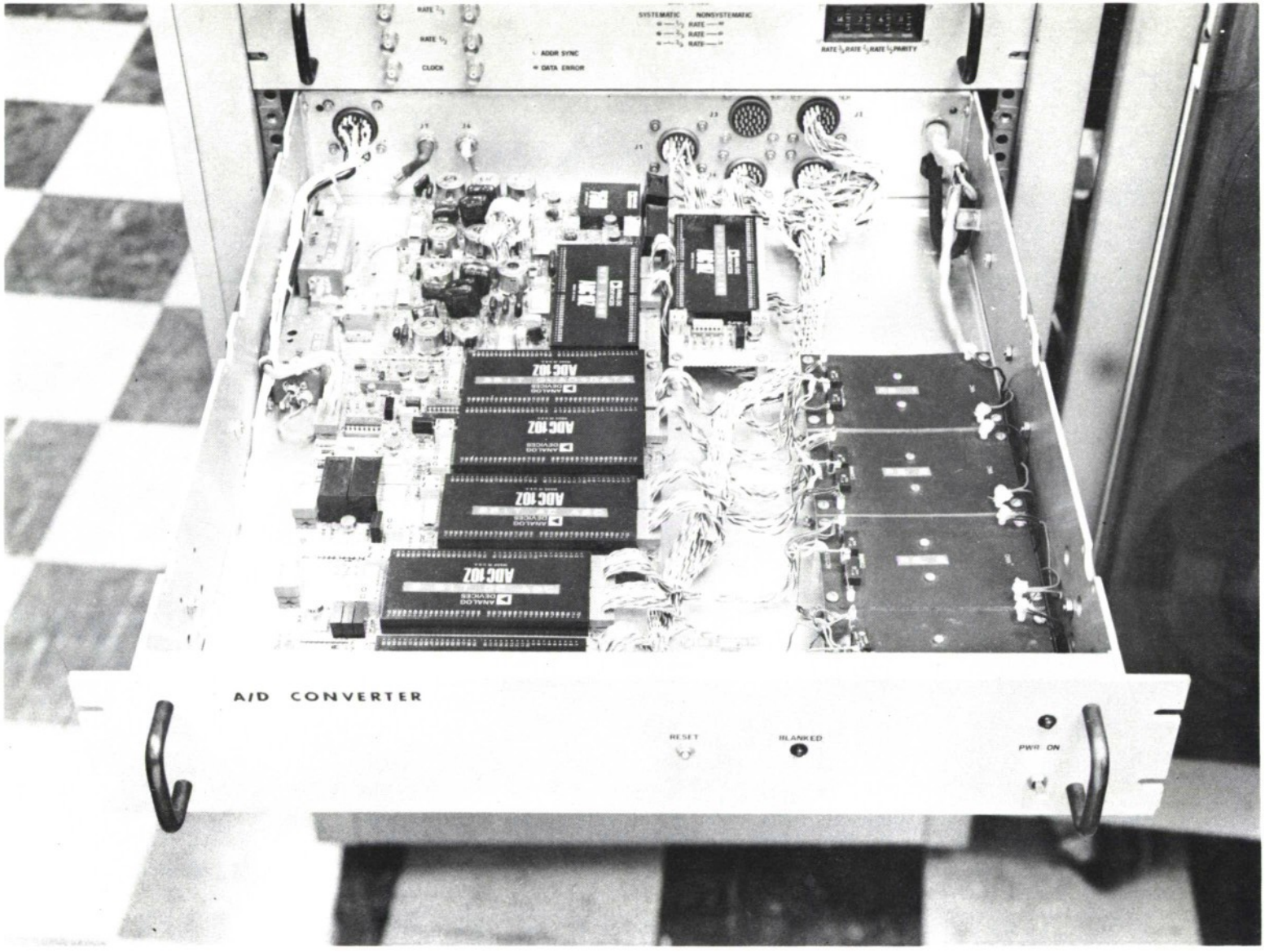


Fig. 25. Analog Drawer.

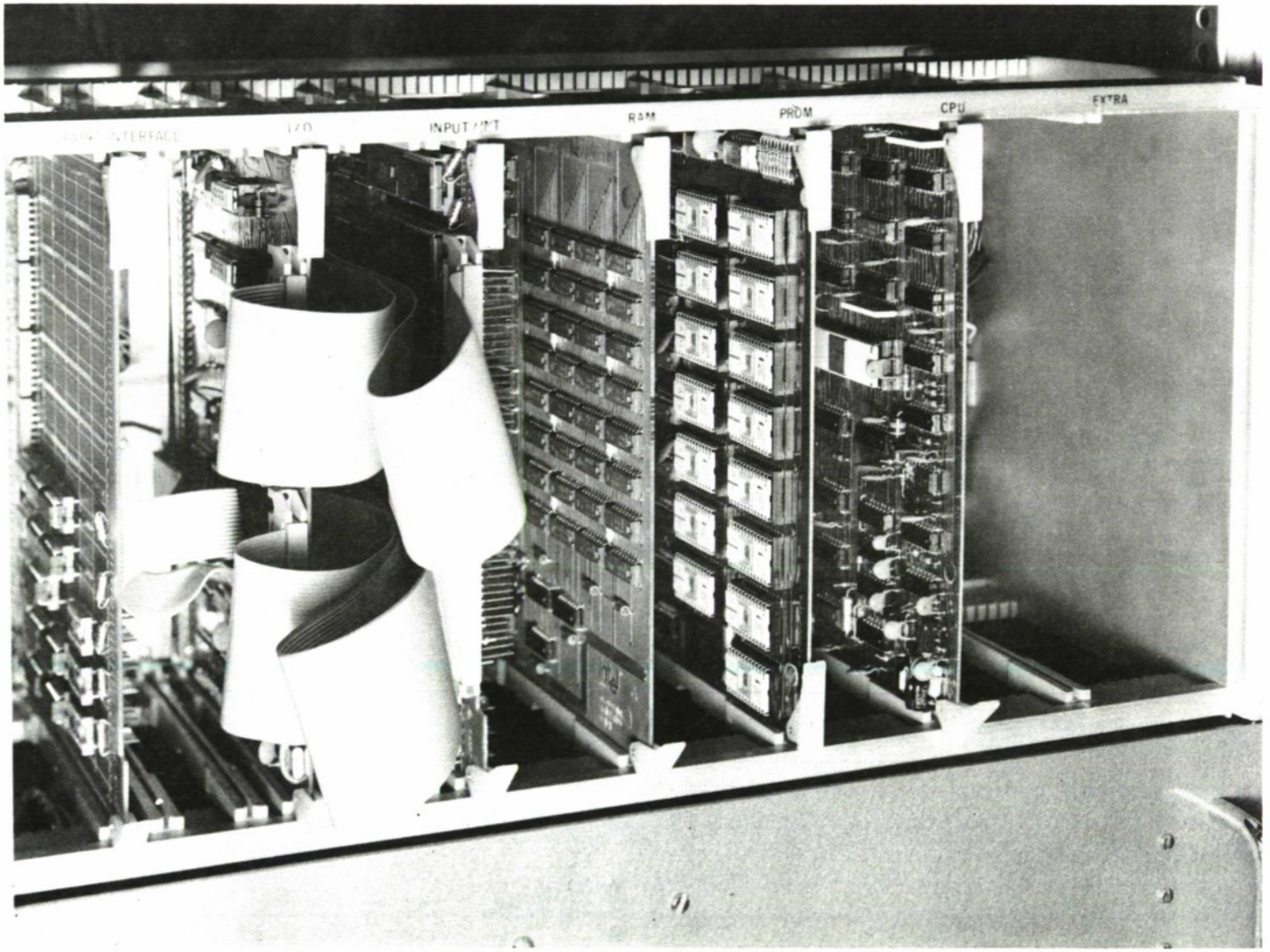


Fig. 26. Controller for Data Recording System.

APPENDIX A
ERROR RATE MEASUREMENT TECHNIQUE

To obtain error rate measurements for non-fading signals and white Gaussian receiver noise we proceeded as follows:

1. The ratio P_r/N is measured with a spectrum analyzer (HP model 8550) at the furthest point into the receiver up to which only frequency translations and linear filtering have been performed. The selected noise bandwidth on the analyzer is 100 Hz (20 dB) and the white noise detector correction is 2.5 dB. The scope display can be read to an accuracy of 1/2 dB. Hence to normalize the signal to noise ratio to a 1 Hz bandwidth we use

$$\frac{(P_r)_{ref}}{N_o} = \frac{(P_r)_{ref}}{N} + 20 - 2.5 = \frac{(P_r)_{ref}}{N} + 17.5 \text{ dB}$$

The ratio is computed for a sinusoidal input signal (no phase modulation) followed by an attenuator set midway in its range A_{ref} · $(P_r)_{ref}$ is the receiver input power and is associated with its attenuator setting A_{ref}. It will be assumed that the attenuator is nearly perfect over the range for which probability of error measurements are to be taken. N_o is the usual one-sided noise spectral density.

2. Next, information in the form of phase modulation is superimposed on the RF input carrier. The energy per chip to N_o ratio is found by dividing by the chip rate of 1200 chips/sec (30.8 dB). Then for any attenuator A setting we have

$$\frac{E_{chip}(A)}{N_o} = \frac{(P_r)_{ref}}{N} - (A - A_{ref}) - 13.3 \text{ dB} \qquad \text{(a-1)}$$

We normally need the energy per information bit to N_o ratio. For the various code rates we have for R=1/2 two chips per information bit, for R=2/3 three chips per two information bits and for R=3/4 four chips per three information bits. Hence the corresponding E_b/N_o ratios would be found from

$$\left(\frac{E_b(A)}{N_o}\right)_R = \frac{(P_r)_{ref}}{N} - (A - A_{ref}) - 13.3 - 10 \log (R) \qquad (a\text{-}2)$$

3. The HP 1645A data analyzer transmitter section provides a pseudo random pattern for up to as many as three parallel data streams which feed into a test multiplexer and DPSK encoder which in turn feeds a binary phase shift modulator to give the test DPSK RF input. At the output of the deinterleaver following the Viterbi decoder we take the decoded estimates of the data streams and feed one of these streams to the receiver section of the analyzer in order to get a direct measurement of the bit error rate.

To obtain error rate measurements for a Rayleigh fading signal with white Gaussian receiver noise we proceed as follow:.

1. A Rayleigh fading simulator was constructed and inserted at the DPSK RF input. The simulator and its calibrator is shown in Fig. A-1. The simulator consists of a voltage controlled attenuator (a) with a control voltage provided by a unit which approximates a Rayleigh fading signal with two choice of fade rates (with equivalent bandwidths of 0.1 and 0.5 Hz). The outputs from two different 31 bit feedback shift registers are low pass filtered to give two independent Gaussian distributions with zero means. Squaring the two outputs and then summing them and taking the square root results in a voltage with a Rayleigh

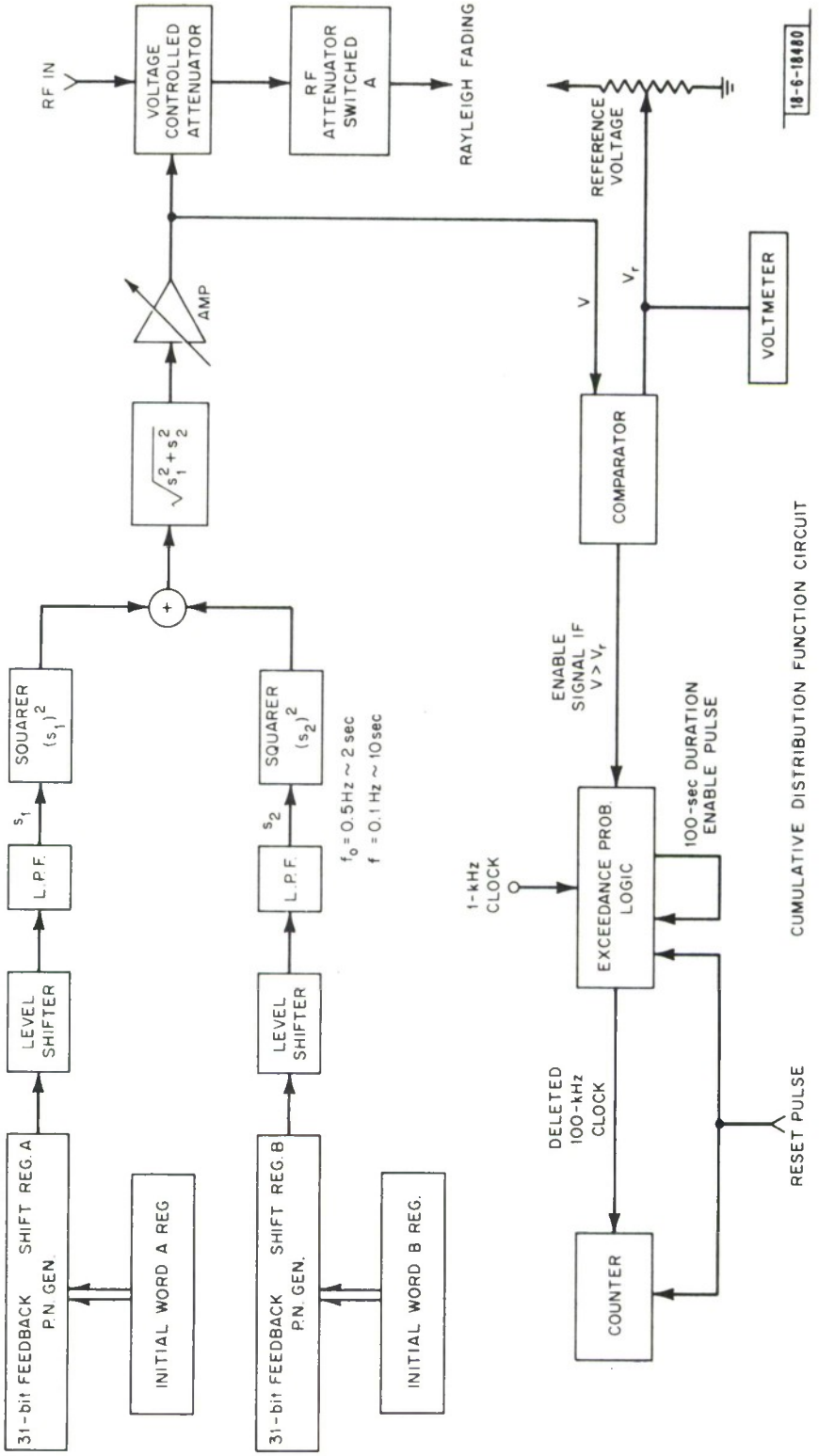


Fig. A-1. Rayleigh Simulator.

characteristic. This can be verified by the cumulative distribution function (CDF) circuit in the following manner. The circuit measures the percentage of time that the input voltage exceeds an adjustable reference voltage (v_r). A 1 kHz pulse train is one input into the exceedance probability logic. For those intervals when $v < v_r$ the pulses are deleted. The number of non deleted pulses in a 100 sec period are counted and divided by 10^5. By varying the reference voltage and utilizing the voltage controlled attenuator characteristic we can produce a plot of the cumulative distribution frunction and compare it to theoretical (Fig. A-2).

2. Once a single point on the error rate curve is determined the RF attenuator A provides the other values of $\bar{E}_b/N_o$ where $\bar{E}_b$ is the average value of energy per information bit. To get the initial calibrated point we note that the attenuator characteristic is approximated by

$$a - a_{ref} = -20(\log(v)-\log(v_{ref}))$$

and at $v_{ref} = V$, $a_{ref} = 0$ dB, therefore

$$p = P_r \left(\frac{v}{V}\right)^2 = C_1 v^2$$

where P is the input power into attenuator a and p is the output power. The probability density function for v is Rayleigh

$$f_v(v) = \frac{v}{\alpha^2} e^{-1/2(v/\alpha)^2}$$

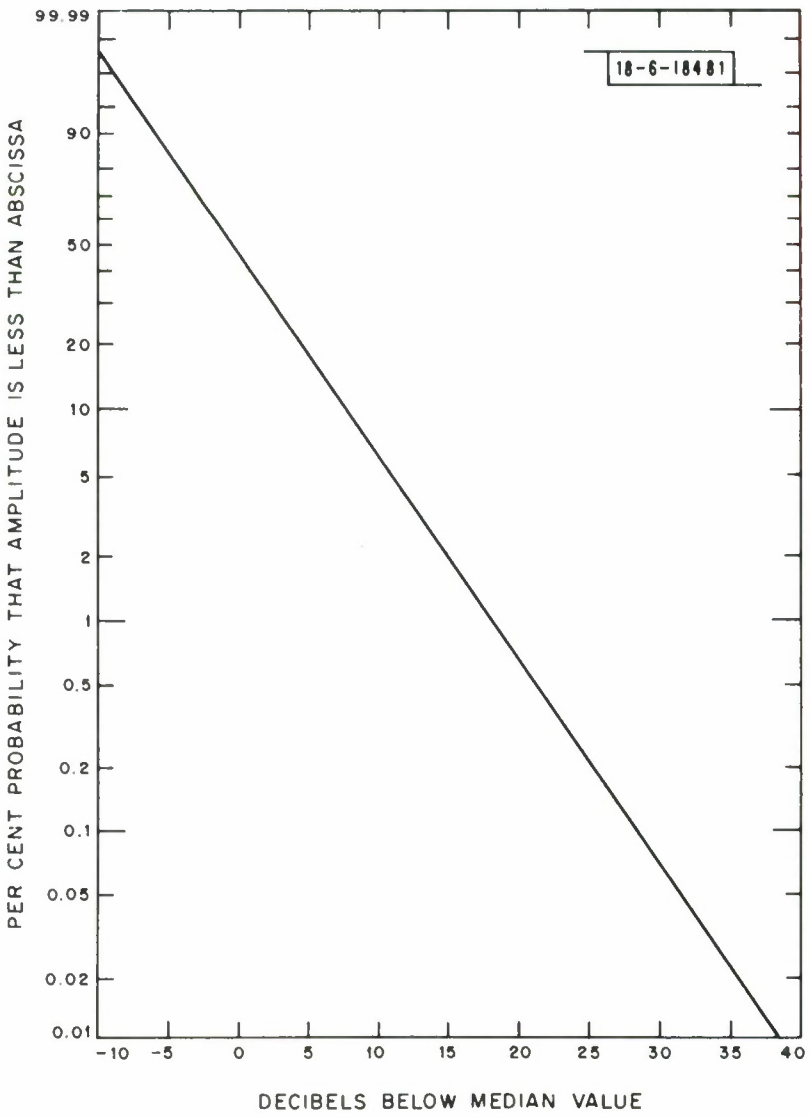


Fig. A-2. Rayleigh probability distribution.

the measured cdf

$$F_v(v_{med.}) = \int_0^{v_{med.}} f_v(v)dv = 0.5.$$

This results in

$$v_{med} = \sqrt{2\ell n2\ \alpha}$$

The probability density function for the output power p is exponentially distributed. This can be shown by a simple probability density function transformation and the result is

$$f_p(p) = \frac{1}{(2C_1^2\alpha^2)e^{-p/2G^2\alpha^2}} \qquad \text{where } C_1^2 = \frac{P}{V^2}$$

The <u>mean</u> attenuator output power is what is desired and this is found to be after substitution

$$\overline{p} = \int_0^{\infty} p\ f_p(p)dp = 2C_1^2\alpha^2 = \frac{1}{\ell n2}\ \frac{P}{V^2}\ v_{med}^2$$

Hence the average attenuation $(\overline{a})_{av}$ with a Rayleigh fading input voltage with median voltage v_{med} is found to be

$$\overline{a}_{av} = 10\ \log \frac{P}{\overline{p}} = 10\ \log \{\ell n2\ (\frac{V}{v_{med}})^2\}$$

With a fixed attenuation values a=0 (i.e. v=V) and $A=A_{ref}$ measure with the spectrum analyzer a reference signal to noise ratio (with no phase modulation) $(P_r/N)_{ref}$. Again by comparison with the nonfading case the energy for chip to N_o raio is then (as a function of the stepped attenuator setting)

$$\frac{\bar{E}_c(A)}{N_o} = \{(\frac{P}{N})_{ref} - \bar{a}_{av} - (A-A_{ref}) - 13.3\} \text{ dB.} \qquad \text{(a-3)}$$

$(P_r/N)_{ref}$ is measured with spectrum analyzer (with Δf = 100 Hz)

A is the stepped attenuator value

$\bar{a}$ = 11 dB for our test configuration.

The energy per bit to N_o ratio for the various code rates is then

$$(\frac{\bar{E}_b(A)}{N_o})_R = \{(\frac{P}{N})_{ref} - 24.3 - (A-A_{ref}) + 10 \log R\} \text{ dB.} \qquad \text{(a-4)}$$

Observe that this method of calibration correctly includes the insertion loss due to the attenuator a.

APPENDIX B

CONSIDERATIONS IN THE DESIGN OF THE DIGTIAL PHASE LOCK LOOP

The SSR-1 receiver has available as an output a 1200 pps stream (developed in the bit synchronizer section) which normally maintains a fixed relation with respect to the output chip stream. It is logical to develop a set of digital system clocks that are exact multiples or submultiples of this 1200 pps stream. In particular, we have selected the Viterbi decoder clock to be 600 KHz and an efficient interleaver design suggests using multiphase 12K, 1.2K and 75 pps clocks. Additionally the data recording system requires clocks which are submultiples of 75 pps. It is also necessary that all these clocks should maintain the same time relationship with respect to the transitions of the SSR-1's 1200 bps output data stream.

A simple solution which provides these clocks is to incorporate a digital phase lock loop in basically a multiplier configuration (see Fig. B-1). The phase detector consists of a few NAND gates and a CMOS D flip-flop whose output varies between the supply voltage (+ 5.0 volts) and ground. The output of the D flip-flop is a pulse train whose pulse width is proportional to the phase difference between the SSR-1 input 1200 pps stream and a 1200 pps stream developed in the (÷ N) circuitry. The level shifter is adjusted such that a 180° phase difference (which gives a square wave input) has a zero mean output. This square wave passes through the filter and the second level shifter is adjusted such that the average output frequency from the VCO is 3 MHz. Note that this particular phase detector will produce a periodic (1/1200 sec) frequency modulation on the 3 MHz even when the two 1200 pps streams maintain an approximate 180° difference. A filter will be selected so that its response at 1200 Hz is

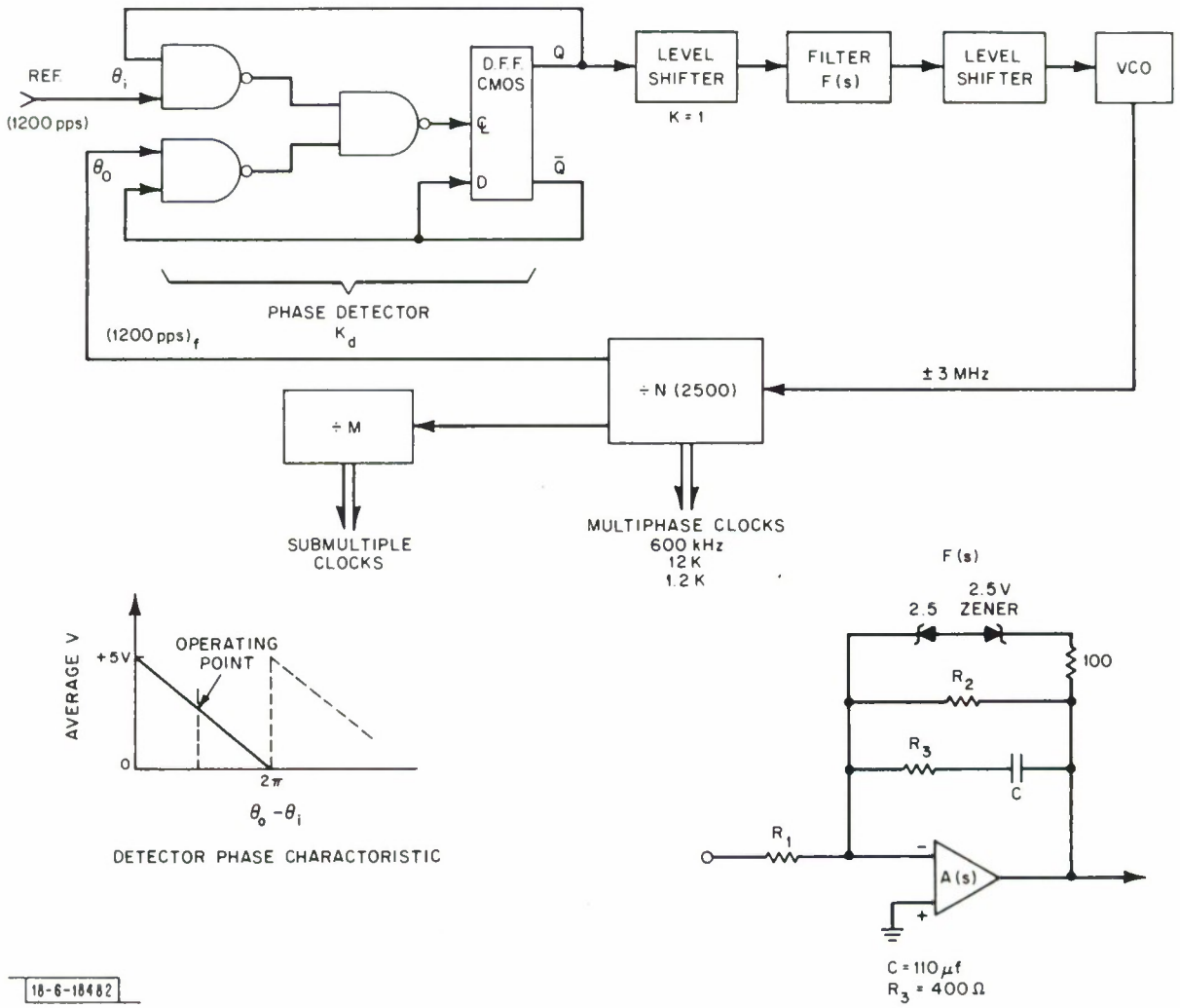


Fig. B-1. Hybrid Phase Locked Loop.

sufficiently low such that modulation results in neglible phase errors on the fed-back signal (i.e., < 1.°). Temporarily ignoring this relatively high frequency FM one can characterize the phase detector by the relationship

$$\nu_{\text{low freq.}} \simeq \frac{5}{2\pi}(\theta_i - \theta_o) \triangleq K_k(\theta_i - \theta_o) \quad .$$

When one takes in account the level shifter the differential transfer relationship (with an assumed 180° difference, 2.5V operating point) given by

$$\delta\nu = K_d\delta\theta$$

The VCO has a transfer characteristic given by $\frac{K_o}{s}$ and the divider a transfer characteristic 1/N. The filter transfer function for large A(s) is given by

$$F(s) = -\left(\frac{R_2}{R_i}\right)\frac{sCR_3 + 1}{sC(R_2 + R_3) + 1} \to -\frac{R_2}{R_1} \text{ as } s \to 0 \quad .$$

This is the type of filter that leads to a second order loop which is the normal filter used for reasons of stability.

The open loop gain is given by

$$G(s) = -\frac{K_o K_d F(s)}{Ns}$$

and hence the closed loop gain by $\left(K_o' \triangleq \frac{K_o}{N}\right)$

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{G(s)}{1 + G(s)} = \frac{R_2(s C R_3 + 1) K_o' K_d}{R_1 C (R_2 + R_3)\left\{s^2 + \frac{K_o' K_d R_2 C R_3 + R_1}{R_1 C (R_2 + R_3)} s + \frac{R_2 K_d K_o'}{R_1 C (R_2 + R_3)}\right\}}$$

Observe that $\lim_{s \to o} \frac{\theta_o}{\theta_i} = 1$ (Fig. b2). Now by comparison with general second-order loop

$$\omega_n^2 = \frac{R_2 K_o K_d}{NR_1 C (R_2 + R_3)}$$

$$2\rho\,\omega_n = \frac{K_o' K_d R_2 C R_3 + R_1}{R_1 C (R_2 + R_3)}$$

Define $\omega_o = \frac{R_2}{R_1} K_o' K_d$, $\omega_3 = \frac{1}{CR_3}$ and with $R_2 >> R_3$, $\omega_2 = \frac{1}{C R_2} \simeq \frac{1}{C (R_r + R_3)}$

Then $\omega_n^2 \simeq \omega_o \omega_2$

$$\rho \simeq \frac{1}{2}\left\{\frac{\omega_n}{\omega_3} + \frac{\omega_2}{\omega_n}\right\} .$$

For $\omega_n >> \omega_2$ $\omega_3 \simeq \frac{\omega_n}{2\rho}$.

The various transfer characteristics are shown in Figs. b-2. For our particular hardware $K_o \simeq 2\pi \times 10^6$ rad/sec volt, N = 2500 therefore

$K_o' K_d \sim 2 \times 10^{+3}$ Hz When tracking a fairly stable pulse train it is normal to want ω_n to be reasonably small and want $\rho \sim .7$ for good error response. Also R_1 must be fairly large so that it draws little current from the CMOS device. From the above equations it is easy to show

$$\frac{R_1}{R_3} = \frac{K_o' K_d}{4\pi \rho f_n}$$

In general we want R_1/R_3 to be very large to get good high frequency rejection (to reduce FM). After consideration we have selected C = 100 μf, $R_3 \sim 400$, $R_2 \sim 100K$ and $R_1 \sim 16K$. This results in $f_2 \sim .016$ Hz, $f_3 \sim 4$ Hz, $f_o \sim 2.0$ KHz, $f_n \sim 5.6$ Hz. Observe that the 1200 pps train out of the phase detector has all its harmonics reduced by

$$\frac{R_2}{R_3} \sim 250 .$$

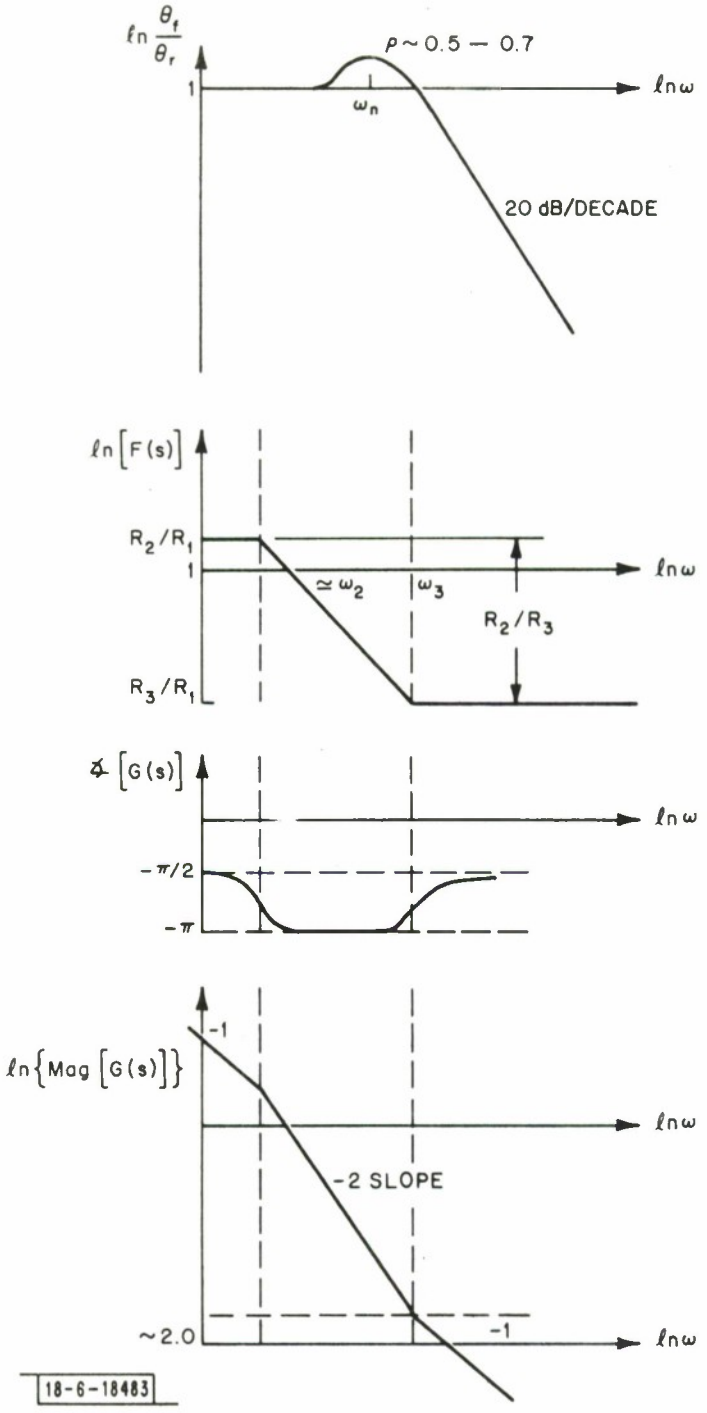


Fig. B-2. Bode Plots for the Ybrid Loop.

The maximum rate of change that can be followed is given by

$$\frac{R_2}{R_3} \simeq 2\pi f_n^2 \sim 200 \text{ Hz/sec} \quad .$$

It should be noted that the largest practical values of C and R_2 results in a minimum value of f_2 which cannot be easily made smaller.

When this PLL configuration was used with the SSR-1 (where we had a constant signal level with Gaussian receiver noise) the P(E) vs E_n/N_o curves came out very close to predicted values. However, when the input level was subjected to the simulated Rayleigh fading we noted that when the signal went into a long fade, the post-decoder error rate would blow up with several hundreds of errors coming up in a matter of minutes. On observation of the SSR-1 bit timing it was then observed that if the signal went away for several seconds the SSR-1 1200 Hz clock which is then free running relative to the absolute data transitions (which are of course determined at the transmitter) often pulls the 1200 Hz clock from the divide-down circuit far enough so that one of the absolute data transitions is crossed. The result is that when demultiplexing the information in, for example, channel 7, it comes out in either channel 6 or 8 and erroneous data continues until frame sync is reestablished. It was also observed that when frame sync is reacquired, the SSR-1 1200 Hz clock would jump quickly and the PLL with its limited bandwidth would not follow appropriately with the result being a lost 1200 Hz pulse. It is obvious that significantly fewer errors would occur if we could flywheel better through this period. To flywheel better, one would have to make f_n considerably smaller. This is not easy to do because of practical limits on the components (in particular R_2, C) and the desire to keep the ratio R_2/R_3 as large as possible to reduce FM on the VCO.

Consequently rather than try to improve the free-run characteristics of the SSR-1 we have adopted a slightly more complicated approach. Figure b-3 shows the system we have added to give better free-run characteristics. The stability is provided by a 1-MHz standard with stability 1×10^{-9}. As long as the frame sync pulses are on, we can slave the 1200 pps clock developed in the master timing chain to the 1200 pps SSR-1 clock. When the frame sync pulse goes away, the 1200 pps clock generated in the timing chain is counted down indirectly from the 1-MHz standard alone.

Basically the circuit works as follows. The two input 1200 pps streams are shaped to give two trains whose trailing edges can differ only by multiples of 1.66 μsec and whose pulse widths are 1.66 μsec. The positive or negative time difference logic produces an output pulse where the pulse width is equal to the difference in time between the trailing edge of the input trains. This positive going pulse also enables a counter which records the number of 1.66 μsec cycles of the pulse duration. The trailing edge of the "difference" pulse then enables a sequencer which transfers the maximum count to a downcounter which in turn is used to provide the slew duration so that on the next 1200 pulse the two trains are put into closer proximity. The last count on the sequencer resets the sequencer as well as the counter. Now if the SSR-1 pulse train increases its rate, then we must add pulses at the 600 KHz rate so that the master divided down stream at 1200 Hz increases as well. This is accomplished by forcing divide 'B' into a (4) mode. Note that output from the divide 'B' circuit also clocks a divide 'C' circuit which is in turn the clock for the slew down-counter. The result is that for every clock pulse into the slew downcounter we have added an extra pulse into the variable 600 KHz line. When the count in

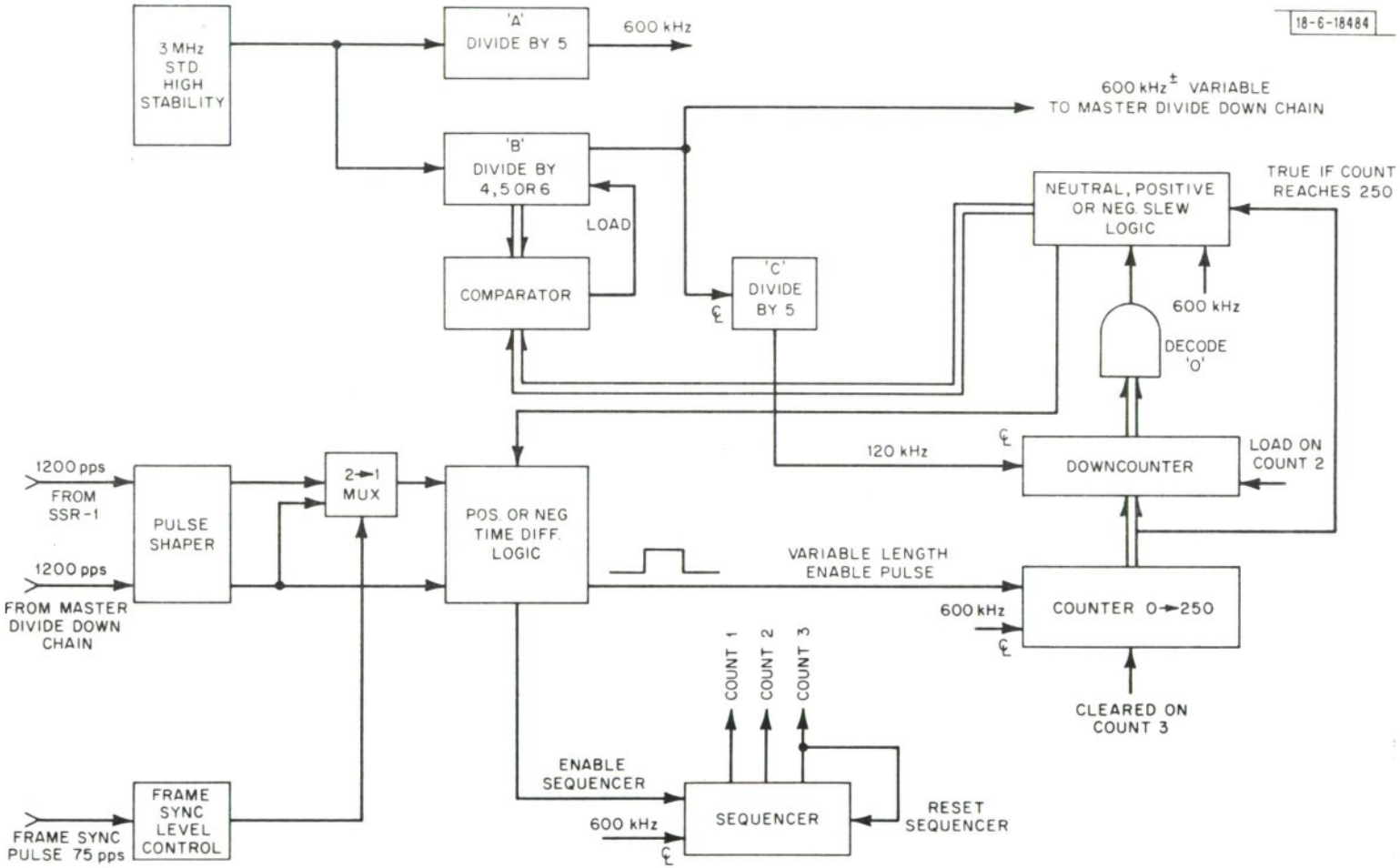


Fig. B-3. "Narrowband" all Digital Phase Locked Loop.

the downcounter reaches '0' the slew is stopped by forcing the divide 'B' counter into the normal (÷ 5) counter. If earlier it had turned out that difference pulse width was going to exceed 1/2 the period, i.e., 433.3 μsec (= 1.66 x (250)) then the neutral, positive or negative slew logic would have detected the count 250, reassigned the roles of which of the 1200 Hz trains was leading and forced the divide 'B' into a ÷ 6 mode for the requisite duration.

The PLL parameters for this PLL configuration are not easy to characterize analytically. However, we have configured the inputs to the downcounter so that it can only slew $\pm$ 1.66 μsec in every 1/1200 sec period. The transmitter clock is a cesium standard with short-term stability better than 10^{-9}. Then as long as our receiver standard for the digital PLL has a short-term stability better than about 10^{-3} we will maintain lock. With the receiver standard in use having a short-term stability of 10^{-9} or better we have no trouble in running through periods of frame sync loss. This system has been checked out extensively and we have yet to observe the catastrophic complete interleaver dump of errors as witnessed with the earlier PLL with wider bandwidth. The hold in range appears to be somewhat less than $\pm$ 2 Hz about 1200 Hz.

APPENDIX C

CONVOLUTIONAL INTERLEAVER DESIGN

The convolutional interleaver can most easily be visualized as a stack of shift registers of different lengths which are examined in sequence (a data bit being written in one end and a bit being simultaneously read from the other). (See Fig. C-1.) The convolutional interleaver has the desirable property that one has control over the minimum time separation between what were (prior to interleaving) "nearby" bits. A deep channel fade may affect up to 150 consecutive bits. The function of the interleaver is to spread the block of erroneous bits over a period of time so that the decoder is influenced by at most a few errors at a time. The errors are really spread out in a periodic manner rather than randomly but the effect is the same on the decoder performance. The desired "spacing" between errors depends on the code structure and for a convolutional code depends on the constraint length K. The design we have chosen separates nearby bits by at least three constraint lengths. The maximum K used is 9 and accordingly we have selected the minimum separation to be 32 bits since this is a power of 2 and is determined by memory utilization considerations.

The selected interleaver structure has rows of delay lines which differ by 4 bits with the largest delay line being 128 bits. The (32 x 128) input bit matrix is overlaid on Fig. C-1. Let us assume that the input bit pattern for a single stream is i_1, i_2, i_3.... . The output interleaved stream is then i_1, i_{130}, i_{259},....etc., where the output stream is subjected to channel degradation. Let us assume that the channel degradation consists of a fade that starts at i_1, and continues up to 4 x 32 = 128 bits. The bits affected are those along the diagonal, i.e.,

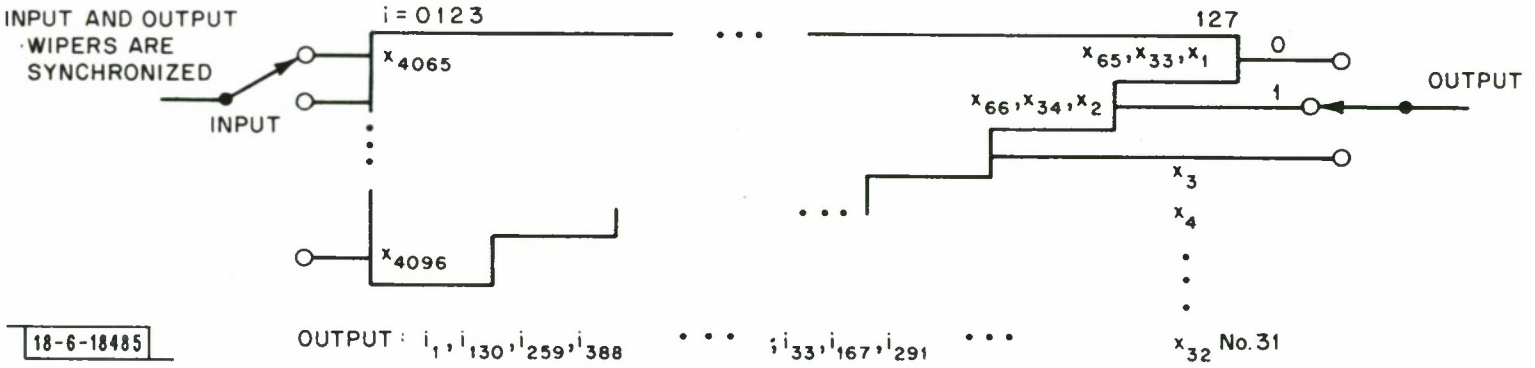


Fig. C-1. Convolutional Interleaver.

$i_1, i_{130}, i_{259}, i_{288}, \ldots\ldots; i_{33}, i_{167}, i_{291}, \ldots; i_{97}, \ldots\ldots i_{4096}$

Note that when deinterleaved and prior to decoding the minimum distance between affected bits is 32. If however more than 128 bits are affected one starts to get consecutive double errors (i.e., 129 and 130, 258 and 259, 387 and 388, etc.). We are then limited to fades of 1.7 sec $(\frac{128}{75})$ at a 75 bps rate . We have selected this size input matrix since it corresponds to a 4K static RAM although we have implemented it with four 1K static RAMs.

One can obtain better protection (i.e., no consecutive errors for 3.4 sec) by doubling the size of memory (a 32 x 256 matrix where the shift registers differ by 8 bits in length). We will restrict our discussion to the prior case but the expansion is obvious.

We have developed a simple realization of the interleaver. The memory will consist of four 1K static RAM's (1024 x 1's). The RAM's are divided up into equal segments of 128 bits as shown in Fig. C-2. We can outline a simple program that describes the interleaver operation as follows:

A. Consider the memory in Fig. C-2 instead as a stack of 128 bit registers (there will then be a matrix of 32 rows and 128 columns).

1. Start at column i (where $0 \leq i \leq 127$) and at row k = 0

a) Read from memory the input bit in column i

b) Write into memory the output bit from column $(i + 127) \cdot \bmod 128$

(Note that in Fig. C-1 for the top row the read and write bit position are separated by 127 positions. The mod 128 takes care of the wrap-around of the memory.)

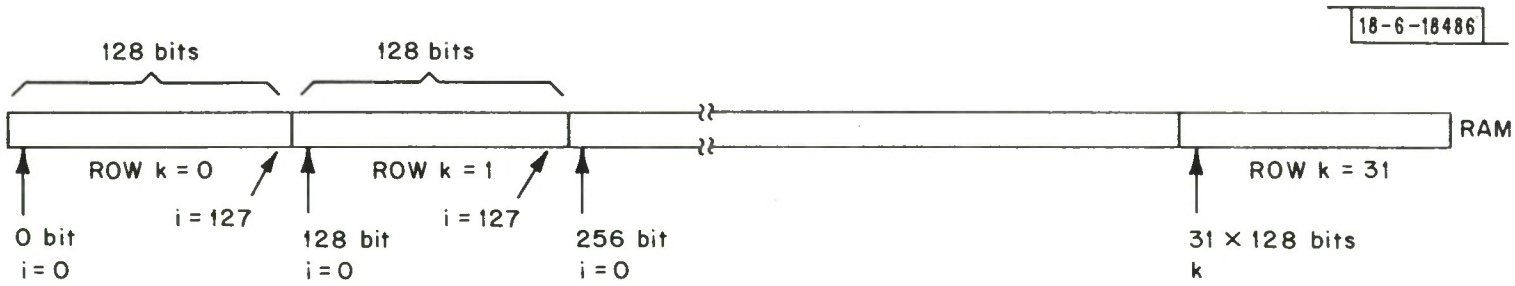


Fig. C-2. Memory Segmentation.

2. Proceed to row k = 1 (with the same i)

a) Read from memory location 128 + i

b) Write into memory location 128 + (i + 127 -4) mod 128

(128 is the origin of the second memory segment)

k) Proceed to row k = j (with same i)

a) Read from memory location 128(j) + i

b) Write into memory location 128(j) + (i + [127 - 4j]) mod 128

etc.

32) Proceed to row k = 31 (with same i)

a) Read from memory location (128) (31) + i

b) Write into memory location (128) (31) + (i + 3) mode 128

B. Next increment i (i → i + 1) and return to A.

The corresponding deinterleaver has the same algorithm except that the general term [127 - 4j] is replaced by [127 - 4$\bar{j}$] where $\bar{j}$ is the 1's complement address.

A realization in hardware of the address program is shown in Fig. C-3 (minus the detailed control of write and read signals for the memory as well as the memory chips themselves). The counters are segmented to give an i counter (with counts 0 to 127) and a k counter (with counts 0 - 31). The key features of the hardware realization take advantage of binary arithmetic. For example for the 12 bit addresses.

a) 128j is realized by tying the k counter to the 5 msb's of the 12 bit address.

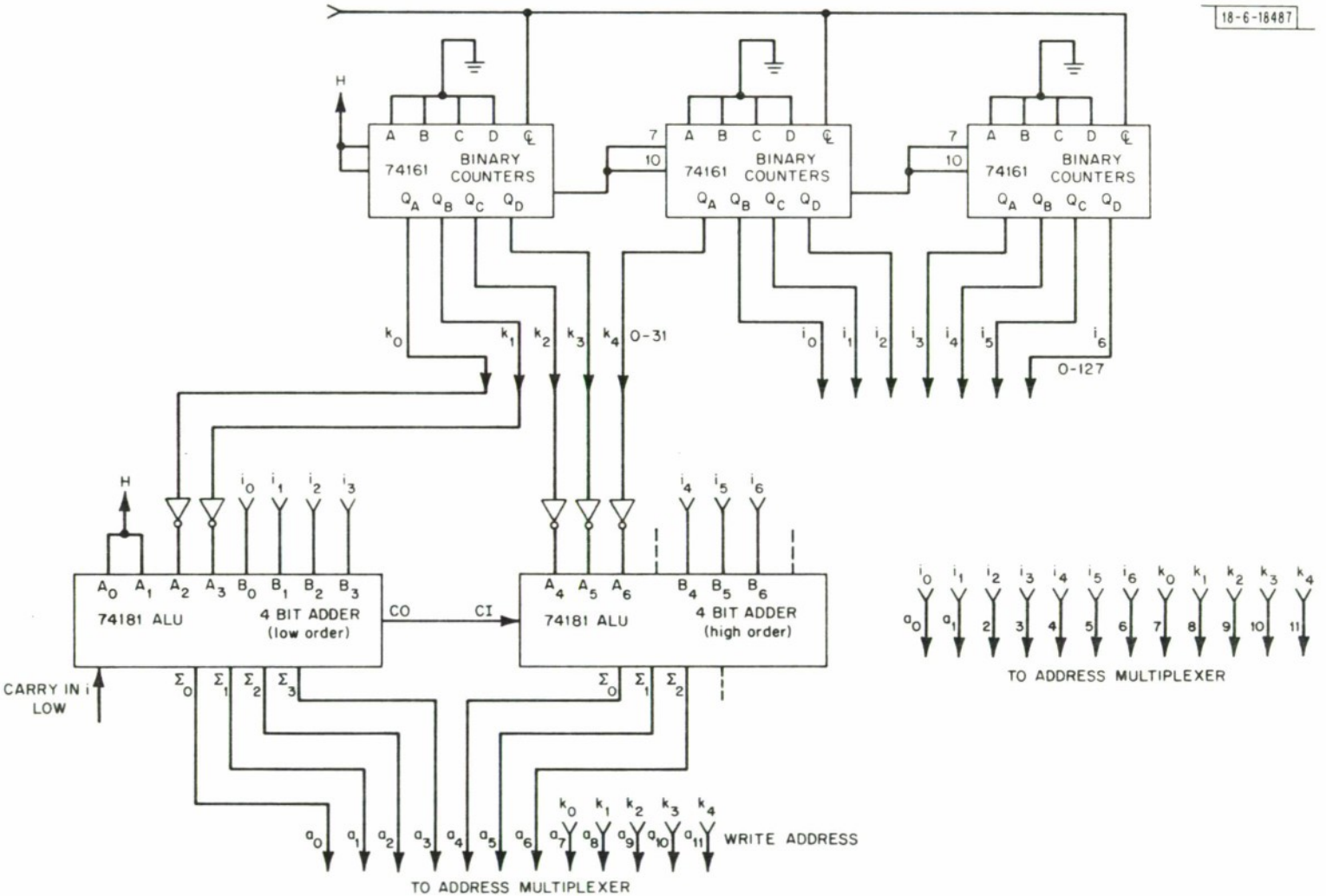
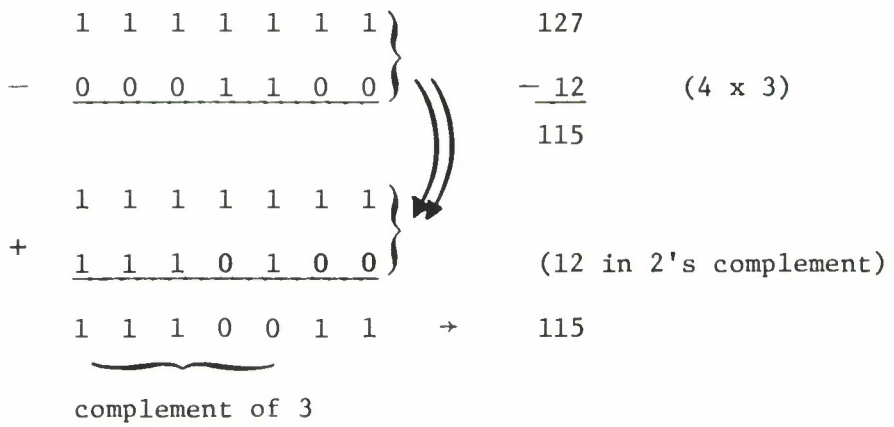


Fig. C-3. Interleaver Implementation (Minus Memory & Memory Control).

b) (i + [127 - 4j]) mod 128 is realized in two simple steps. [127 - 4j] is realized without any computation since for example



Note that the 5 most significant bits are complements of the k counter. Hence 127 - 4j is implemented with A_0 and A_1 tied high and A_2 to A_6 being the inverted output of the k counter.

Next the addition is formed and the mod 128 is performed by ignoring A_7 and A_8.

Obviously the result (i + [127 - 4j]) mod 128 is tied to the 7 least significant bits of the interleaver.

The binary counter will be common to all interleavers, deinterleavers as well as the large compensating delay lines. The additional chips need for each interleaver is a few 4-bit adders and a multiplexer plus a control gate for switching between read and write addresses.

ACKNOWLEDGEMENTS

We gratefully acknowledge the active support of the Naval CAMS teams on Hawaii and Guam during the difficult period when the Fleet Broadcast was becoming operational. Lt. Commander George Burman should be singled out for his support both while in Washington and Guam.

The author would like to thank E. Bucher for his help on some of the problems that arose during the implementation of the experiment. E. Driver deserves much of the credit for the analog circuit design and F. Klatt the credit for the organization of the data collection facility.

REFERENCES

1. R. K. Crane, "Morphology of Ionospheric Scintillation," Technical Note 1974-29, Lincoln Laboratory, M.I.T. (21 May 1974) DDC AD-780552/9.

2. E. A. Bucher, "UHF Satellite Communication During Scintillation," Technical Note 1975-10, Lincoln Laboratory, M.I.T. (5 August 1975) DDC AD-A014801/5.

3. J. A. Heller and I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communications," IEEE Trans. Communications Tech. Com-19, No. 5 (1971).

4. A. J. Viterbi, "Convolutional Codes and Their Performance in Communications Systems," IEEE Trans. Communications Tech. Com-19, No. 5 (1971).

5. W. C. Lindsay and M. K. Simon, Telecommunication System Engineering, (Prentice Hall, New Jersey, 1973).

6. R. K. Crane, "Ionospheric Scintillation," Proc. IEEE 65, No. 5 (February 1977).

7. J. Odenwalder, "Optimal Decoding of Convolutional Codes," PhD Thesis, Dept. of Electrical Engineering, UCLA (1970).

8. E. Paaske, "Short Binary Convolutional Codes with Maximal Free Distance for Rate 2/3 and 3/4," IEEE Trans. Information Theory, IT-20, (September 1974).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-77-217	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* A Time Diversity Coding Experiment for a UHF/VHF Satellite Channel with Scintillation: Equipment Description		5. TYPE OF REPORT & PERIOD COVERED Technical Note 6. PERFORMING ORG. REPORT NUMBER Technical Note 1977-22
7. AUTHOR*(s)* David P. White		8. CONTRACT OR GRANT NUMBER*(s)* F19628-76-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element No. 33109N
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Electronic Systems Command Department of the Navy Washington, DC 20360		12. REPORT DATE 1 September 1977 13. NUMBER OF PAGES 106
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* Electronic Systems Division Hanscom AFB Bedford, MA 01731		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

satellite communications
scintillation
UHF (ultra high frequency)
VHF (very high frequency)
equatorial regions
code rates
gapfiller satellite

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Satellite communications in the UHF and VHF bands has at various times and in certain receiver locations on the earth's surface been subject to moderate to severe degradation as a result of scintillation. The affected receiver areas are confined primarily to the equatorial regions and to a lesser extent over the sub-auroral to polar cap areas. For the equatorial region (more specifically, a band centered on the geomagnetic equator) the scintillation is largely a nighttime phenomenon beginning shortly after local sunrise and diminishing after midnight. In response to the potential problem on the Navy's UHF fleet broadcast channel we have conducted a study on the communication alternatives as well as an extensive field experiment. The field experiment was conducted between Hawaii and Guam over the Pacific Gapfiller (Marisat) satellite from August 1976 through January 1977.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*